UNIVERSITY OF CALIFORNIA
SANTA CRUZ

A COMPREHENSIVE ANALYSIS OF TRACKING AS A DATA ASSOCIATION PROBLEM

A dissertation submitted in partial satisfaction of the
requirements for the degree of

Doctor of Philosophy

in

COMPUTER SCIENCE

by

Eric C. Poblenz

December 2015

The Dissertation of Eric C. Poblenz
is approved:

Professor Gabriel Elkaim, Chair

Professor Dimitris Achlioptas

Professor Bruce Sawhill

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Contents

List of Figures

List of Tables

Abstract

A Comprehensive Analysis of Tracking as a Data Association Problem

by

Eric C. Poblenz

Algorithms based on traditional notion of tracking as a state estimation problem yield just a single interpretation of the data. For some applications, the ability to identify ambiguities and compare different interpretations using a well-defined measure of confidence is critical. Such applications require a direct solution to the data association problem in order to characterize the relevant uncertainty. This notion of tracking has received relatively little attention largely due to a failure to recognize its utility beyond maintaining the state estimation process. As a result, the options available to the practitioner are limited and the performance of statistical data association models is not well understood, especially in terms of the quality of the sample they produce.

This work has sought to change that by developing a new data association model that extends the scope and flexibility of existing models. The questions of how to specify an objective prior distribution over data association hypotheses and how to efficiently perform inference on the high-dimensional posterior distribution are very much open. To help provide answers, we considered numerous different priors, including Bayesian nonparametric models and several models never before applied to tracking. With regard to inference, we considered various implementations of Markov chain Monte Carlo (MCMC) and population Monte Carlo (PMC) samplers. A comprehensive evaluation was performed in the context of a wide-area radar surveillance application.

This thesis is dedicated to my wife, Gisela, whose love, patience, and unwavering support saw me through to the end despite more obstacles than we could have possibly imagined.

Chapter 1

Introduction

Over the years, target tracking has been the subject of a considerable amount of research because of its importance in applications in areas such as defense, astronomy, medical imaging, robotics, and air traffic control. The various approaches described in the literature cover a multitude of different target types, dynamics models, sensors, and operating environments. Many of these approaches have led to effective algorithms that have been successfully applied to real-world problems. However, despite a wealth of developments, there are still tracking applications that are not well-served by any existing technique.

Beginning with the earliest attempts at using digital computers to estimate the velocity of a target based on a series of positions from radar contacts, tracking algorithm research has focused primarily on the notion of tracking as a kinematic state estimation problem. The objective in this case is to follow the target, integrating the information obtained from a sequence of noisy measurements, in order to minimize the uncertainty in our knowledge of its state. The resulting estimate has obvious utility in predicting the future state of the target. As the dominant paradigm, there exists a rich set of solutions related to state estimation.

Alternatively, tracking can be viewed as a data association problem. The objective in this case is to process the information obtained from the measurements in order to minimize the uncertainty in our knowledge of their origin. In other words, the objective is to determine the correct associations between the measurements and the targets that generated them. Here, the resulting estimate is useful for making inferences and, most importantly, quantifying the uncertainty with regard to things like the number of targets, track identity, and target crossing patterns. This perspective on tracking has received relatively little attention from researchers in part because data association, despite being a well-studied concept in the field, is usually treated as a subproblem of state estimation.

Data association and state estimation are indeed intimately related. The uncertainty in the measurement origins must be addressed, in some respect, when generating a state estimate. Likewise, the uncertainty in each target's state must be addressed, in some respect, when generating an estimate of the measurement origins. However, their respective roles differ depending on how the tracking problem is posed.

Algorithms based on the notion of tracking as a data association problem distinguish themselves from those designed to produce state estimates through a different

treatment of the relevant uncertainties in their models. The difference can be quite important for certain applications, such as surveillance, but is often overlooked.

The goals of this dissertation are to emphasize the distinctions between the two notions of tracking and to provide a comprehensive analysis of the latter, which we shall refer to as *associative tracking*. In the process, we will introduce a new associative tracking model that extends the scope and flexibility of existing models. The proposed model will be evaluated on the difficult, and increasingly important, problem of tracking people using radar in a wide-area persistent surveillance application.

1.1 Motivation

Wide-area surveillance in order to protect vital assets such as borders, ports, pipelines, refineries, military bases, or power plants from terrorists attacks and other threats is rapidly becoming an important issue worldwide. There is a growing need for the persistent monitoring of large outdoor areas in both the defense and commercial industry.

Traditionally, the primary sensor technologies for such surveillance were electro-optical (EO) and infra-red (IR) cameras. Although cameras can provide sufficient resolution for target identification, their limited field of view makes it more difficult to rapidly scan large areas to identify small targets. In addition, adverse weather and atmospheric conditions can significantly degrade the performance of EO/IR sensors . More recently, radars have become the go-to sensor in many wide-area surveillance applications because of their ability to scan large areas quickly and operate in all weather and lighting conditions. Improvements in detection capabilities for small targets, even at long ranges, and lower costs has made them feasible for use in an increasing number of applications.

An example of one such application comes from the Secure Border Initiative (SBI) established by the U.S. Department of Homeland Security (DHS) back in 2006. Through its SBI*net* program, the initiative sought to deploy the technology necessary to monitor the nearly 6000 miles of northern and southern land borders of the United States. The backbone of the system was a network of tower-mounted radars designed to find and track human targets. Once the radar had acquired a target, higher resolution sensors (such as EO/IR) would be cued to help identify and classify it. Although the

program was ultimately canceled in 2011 because of concerns about cost and effectiveness [150, 33], the need for this kind of surveillance capability remains. The experience validated the use of stationary radar for wide-area ground surveillance, but exposed some of the limits of conventional tracking algorithms when applied to this type of problem.

Another example is the U.S. Army's need for persistent surveillance around forward operating bases in order to provide critical situational awareness. For this purpose, they currently employ the Persistent Threat Detection System (PTDS) and its smaller cousin the Persistent Ground Surveillance System (PGSS) [192]. These are comprised of aerostats (i.e. unmanned unpowered balloons), tethered to ground control stations, that carry a suite of sensors and communications equipment. Such systems are an attractive lower cost alternative to flying unmanned aerial vehicles (UAVs) to perform the same mission. Early generations relied primary on EO/IR sensors, or used radar exclusively for detecting low-altitude aircraft and vehicles, but the latest incarnations include modern radar technologies that enable the detection of "dismounted forces".

The use of a stationary radar to continuously scan vast outdoor regions in order to find and track human targets is indeed an important problem. On the surface this is a classic multiple target tracking (MTT) problem with application-specific choices of sensor, target, and environment. However, a closer examination reveals that the underlying assumptions made in conventional approaches to MTT are not well suited to the particular requirements of this application.

Conventional approaches frame tracking as a state estimation problem. This leads to algorithms whose output is a set of descriptions, one for each target, of the distributions over the possible states. When the state distributions are Gaussian, each description consists of the sufficient statistics: mean and covariance. For general state models that do not have an analytic form, each description consists of a set of samples from which various properties of the distribution can be estimated.

The problem with this is that it does not characterize the uncertainty that is most important in a surveillance application. The primary objective in this case is to understand what is happening within the surveillance region and it is this uncertainty, in terms of the number of targets and their trajectories, that needs to be quantified. While the output of a conventional tracker does provide a distributional estimate of the state of each target, it only provides a point estimate of the overall picture. It conveys

just a single estimate of the number of targets and a single set of state distributions which describe their respective trajectories. The state distributions themselves are derived from either the best estimate of the mapping of measurements to targets or an approximate marginalization over all possible mappings.

Despite the fact that the uncertainty in the individual trajectories themselves is modeled and quantified, the state distributions are estimated using one interpretation of the data. Even if that interpretation is optimal in some sense, the output only offers one view of what is happening and does not tell us anything about the alternatives. This poses the risk that a threat or target of interest might be missed.

Conventional tracking algorithms treat the mapping of measurements to targets as a subproblem that needs to be solved in order to estimate the distributions over states for each target. This is appropriate when the primary objective is to obtain state estimates and the distribution over possible mappings can be assumed to have one dominant mode so that it need not be subject to any further analysis. In general, this is the case for applications in which there are relatively few false alarms or there exists a good physical model of the target dynamics. In such applications, a detection is a pretty reliable indicator that a target is present and the question of which measurements are associated with which targets is only of interest to ensure that the state estimation for each target is based on coherent input.

Most existing tracking algorithms were motivated by applications involving airborne targets that share the aforementioned traits. For aircraft, missiles, and projectiles we have excellent dynamics models that enable us to accurately predict the future state of such targets given a good initial estimate. The objective of tracking here is to obtain such estimates from the noisy measurements in order to facilitate predictions. Except for extreme scenarios, the false alarm environment is such that the distribution over mappings of measurements to targets is sufficiently peaked that averaging over multiple mappings, or simply using the best guess, can be expected to yield a reasonable state estimate.

For the present application, however, a better approach is to frame tracking as a data association problem. From this perspective, state estimation becomes a subproblem that needs to be solved in order to estimate the distribution over mappings of measurements to targets.

The mappings are the focus here because they in fact provide a picture of what

is happening within the surveillance region. Each mapping of measurements to targets represents a set of beliefs about the origin of each measurement. Note that the mappings are not necessarily one-to-one as some measurements may originate from noise sources and be regarded as false alarms, while others may cover multiple targets. Each mapping also tells us the number of targets, where they are, and how they are moving.

Clearly, if we limit our consideration to the measured quantities referenced by the mapping, then our knowledge of the where and how the targets are moving will be somewhat imprecise (although sufficient in some instances) and perhaps incomplete. Fortunately this is not necessary. Since the mapping tells us which measurements originated from a each target, we can readily derive more precise estimates for the respective trajectories using state estimation. In other words, there is an equivalence between a mapping of measurements to targets and a set of state distributions.

This approach leads to tracking algorithms whose output is a description of the distribution over mappings of measurements to targets. Since this distribution has no analytic form, the description will consist of a set of samples. These samples can then be analyzed in order to characterize the uncertainty in different interpretations of the data and hence different views of what is happening in the surveillance region. Thus, the output provides a distributional estimate that is directly matched to the primary objective.

Having such a distributional estimate enables us to visualize the uncertainty for a human operator or quantify it for further automated analysis. The samples can also be analyzed to find the modes of the distribution or to identify and quantify things like ambiguities in target identity or crossing patterns.

Of course, state estimation does not cease to be important even when tracking is framed as a data association problem. In fact, it can be regarded as the secondary objective in this case. Although it was treated here as a subproblem in the process of generating the output, once a particular mapping is deemed to be of interest we can appropriately shift focus to state estimation and use the mapping to generate any or all of the state distributions.

1.2 Background

1.2.1 State Estimation

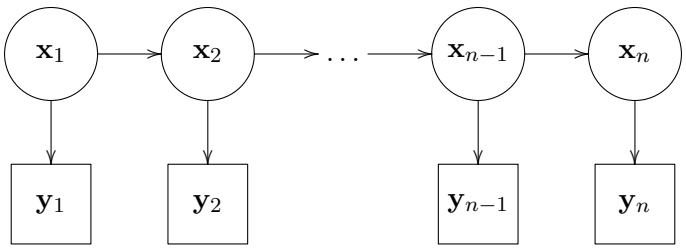
Whether as the primary or secondary objective – depending on how the tracking problem is posed – kinematic state estimation plays a vital role in all tracking systems. The process can be described as follows.

For a given target, let its state at time n be represented by the vector $\mathbf{x}_n$. The state of the target is assumed to be unobserved and evolve over time as a Markov process. With each scan of the sensor, the corresponding latent state is assumed to emit an observable measurement. This implies that conditional on the previous state, the target's state at time n is assumed to be independent of all earlier states and assumed to probabilistically generate the measurement vector $\mathbf{y}_n$. These assumptions define a generative model in which the joint distribution over states and measurements has the form:

$$\begin{aligned} p(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{y}_1, \ldots, \mathbf{y}_n) &= p(\mathbf{y}_1, \ldots, \mathbf{y}_n \mid \mathbf{x}_1, \ldots, \mathbf{x}_n)\, p(\mathbf{x}_1, \ldots, \mathbf{x}_n) \\ &= p(\mathbf{y}_1 \mid \mathbf{x}_1)\, p(\mathbf{x}_1) \prod_{i=2}^{n} p(\mathbf{y}_i \mid \mathbf{x}_i)\, p(\mathbf{x}_i \mid \mathbf{x}_{i-1}). \quad (1.1) \end{aligned}$$

The target dynamics are modeled by the state transition distribution, $p(\mathbf{x}_i \mid \mathbf{x}_{i-1})$, and the measurement process is modeled by the emission distribution, $p(\mathbf{y}_i \mid \mathbf{x}_i)$. Note, however, that the state of a target is not limited to strictly representing its kinematic properties such as position, velocity, or acceleration. To satisfy the Markov property, the state need only provide a complete summary of the target's past. This may include any relevant time-varying parameters. Note also that the state transition distribution and emission distribution may be time-varying.

This factorization of the joint distribution over states and measurements is equivalent to the following graphical model:



where the circles represent the latent state variables and the squares represent the

observed measurement variables. The conditional independence structure expressed here is that of a standard Hidden Markov Model (HMM).

The goal here is to estimate, for each target, the posterior distribution of the state variables given the measurements: $p(\mathbf{x}_1, \ldots, \mathbf{x}_n \mid \mathbf{y}_1, \ldots, \mathbf{y}_n)$. However, to reduce computation and storage requirements, rather than computing the full joint posterior, it is common to sequentially estimate the marginal posterior of the state at the current time step. This is known as the filtering distribution which is given by:

$$p(\mathbf{x}_i \mid \mathbf{y}_1, \ldots, \mathbf{y}_i) = \frac{p(\mathbf{y}_i \mid \mathbf{x}_i)\, p(\mathbf{x}_i \mid \mathbf{y}_1, \ldots, \mathbf{y}_{i-1})}{p(\mathbf{y}_i \mid \mathbf{y}_1, \ldots, \mathbf{y}_{i-1})}, \tag{1.2}$$

where $i = 1..n$ with:

$$p(\mathbf{x}_i \mid \mathbf{y}_1, \ldots, \mathbf{y}_{i-1}) = \int p(\mathbf{x}_i \mid \mathbf{x}_{i-1})\, p(\mathbf{x}_{i-1} \mid \mathbf{y}_1, \ldots, \mathbf{y}_{i-1})\, d\mathbf{x}_{i-1} \tag{1.3}$$

and

$$p(\mathbf{y}_i \mid \mathbf{y}_1, \ldots, \mathbf{y}_{i-1}) = \int p(\mathbf{y}_i \mid \mathbf{x}_i)\, p(\mathbf{x}_i \mid \mathbf{y}_1, \ldots, \mathbf{y}_{i-1})\, d\mathbf{x}_i. \tag{1.4}$$

When the state and measurement variables are Gaussian, and the dynamics and measurement models are both linear, the (discrete-time) Kalman filter (KF) provides an efficient and optimal recursive solution to the problem of estimating the marginal posterior [95, 6, 13]. The filter is optimal in the sense that, given accurate dynamics and measurement models, the resulting state estimate can be shown to minimize the mean squared error. Even when the variables are not Gaussian, but we have the first two moments, the Kalman filter can be shown to be the best linear state estimator.

Although the Kalman filter does not estimate the full joint posterior, given the sequence of state estimates up to time n, the earlier estimates can be revised to incorporate information from the latter measurements using one of the so-called "Kalman smoother" algorithms, such as the Rauch–Tung–Striebel (RTS) smoother [159]. That is, the smoother processes the estimates $\mathbf{x}_i, \ldots, \mathbf{x}_n$ to produce an estimate of the marginal posterior: $p(\mathbf{x}_i \mid \mathbf{y}_1, \ldots, \mathbf{y}_n)$, where $1 \leq i < n$.

Since its introduction, the Kalman filter has played a central role in many tracking algorithms due to its ease of formulation, suitability for real-time applications, and optimality properties. In practice, it often performs surprisingly well even when the underlying assumptions are not entirely justified making it a good place to start even for complex problems. The standard Kalman filter approach has also been adapted to

accommodate nonlinear dynamic and measurement models. Unfortunately, these new estimators are nonoptimal.

One such adaptation is the extended Kalman filter (EKF), which works by linearizing a nonlinear, but differentiable, function around the current state estimate using a multivariate Taylor series expansion. The Jacobian matrix then takes the place of the linear transformation in the standard Kalman filter equations. Although often used owing to its relative simplicity, the EKF is sensitive to the initial state estimate and modeling errors. In cases where the linearized approximation is poor, the resulting biases can cause the filter to rapidly diverge leading to unbounded estimation errors. Unfortunately, it can be difficult to determine precisely when divergence may occur because it depends on the nonlinear function, the current state estimate, and the magnitude of the covariance. Filter consistency is also a concern as the EKF has been found to have a tendency to underestimate the state covariance.

A more recent adaptation intended to address the limitations of the EKF is the unscented Kalman filter (UKF) [91, 92]. It works by using the unscented transform (UT) to propagate the mean and covariance of the current state estimate through the nonlinear function. The basic idea is to generate a Gaussian approximation of transformed state rather than attempt to approximate an arbitrary nonlinear function. The unscented transform is a deterministic sampling technique in which a minimal set of so-called sigma point points, carefully chosen based on the moments of the input distribution, is transformed by the nonlinear function; the transformed points are then used to recover the moments of the output distribution. For the Gaussian random variables used in the UKF, the mean and covariance estimates are accurate to the 3rd order (of a Taylor expansion) for any nonlinearity [193]. In addition to providing better accuracy than the EKF with computational complexity of the same order, the UKF has the advantage of not requiring any derivatives of the nonlinear functions since they need only be evaluated at the sigma points. Not only does this spare the cost of computing the Jacobian for differentiable models, it allows the UKF to accommodate any dynamics or measurement model including those that are discontinuous or inherently discrete.

For nonlinear or non-Gaussian models, there are no general analytic solutions for optimal state estimation. In cases where the EKF and UKF are inadequate, particle filters have become the method of choice for deriving approximate solutions [71, 47]. Particle filters are a broad class of sequential Monte Carlo (SMC) algorithms that use a

set of randomly selected sample elements (the "particles"), and their associated weights, to represent an arbitrary probability density. The sample elements and weights are generated using sequential importance sampling (SIS). As the number of sample elements increases, this empirical estimate of the density becomes more exact. The moments or even a functional estimate of the density can be obtained directly from the sample. Particle filters can be used to sequentially approximate either the full joint state posterior or the marginal state posterior, at each time step. This flexibility comes with the cost of increased computational complexity that grows with the number of particles. In addition, their performance depends on a having suitable importance distribution, from which the sample elements are drawn at each time step, and this can be difficult to specify for some applications. The difficulty lies in the facts that the importance distribution must approximate $p(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_1, \ldots, \mathbf{y}_n)$, in order to minimize the variance of the weights and hence the variance of the state estimates, and still be a distribution from which sampling is feasible. Even with a good importance distribution, the variance of the weights increases over time so an additional challenge is avoiding too many degenerate particles (i.e. those with weights close to zero). This is usually addressed using some sort of resampling procedure in which particles with negligible weights are replaced by new particles generated close to particles with higher weights [84]. Currently, there is no universal approach to mitigating the impacts of particle degeneracy that works for all applications.

One approach to modeling a richer set of target motions is to consider the distribution, $p(\mathbf{x}_i \mid \mathbf{x}_{i-1}, \mathbf{u}_{i-1})$, where $\mathbf{u}_{i-1}$ is an unknown input that must be estimated along with the state. Another approach is to assume that the target's motion can be described using one of a finite number of different models. The state is then estimated using each of the possible models and either the estimate from the best model is selected or a new estimate is computed as a weighted combination of the possible models. Typically, the latter is preferred and the weighting is based on the posterior probability of the respective model being correct, given the measurements up to the current time. This approximates the state transition model using a finite mixture model. Unfortunately, the optimal implementation is impractical since it requires summing over an exponentially increasing (over time) number of model sequences so nonoptimal approximations are required. For linear-Gaussian models, the standard multiple model approaches are the generalized pseudo-Bayesian (GPB) and the interacting multiple model (IMM) esti-

mator [6]. These approaches merge model sequences that differ in their older steps and approximate the mixture distribution using its first two moments. When the number and structure of the models varies over time, the variable structure estimators can be used [115].

1.2.2 Data Association

When discussing state estimation, we implicitly assumed that it was known which measurements originated from the given target and hence which should be used by the model in (1.1). In general, this will not be the case because the presence of multiple targets, missed detections, or false alarms all create uncertainty in our knowledge of the measurement origins. The data association problem is to resolve this uncertainty by determining the correct mapping of measurements to their sources. Given this mapping, which may be represented as a probability distribution to quantify the uncertainty, state estimation can be performed. Of course, when tracking is posed as a data association problem, obtaining this mapping is not just a subproblem to be solved for the sake of state estimation but the goal itself.

Let $\omega = \{\tau_0, \tau_1, \ldots, \tau_K\}$ denote the mapping of measurements to their sources,[1] where τ_0 is the set of indices to the false alarm measurements and τ_k is the set of indices to the measurements originating from the k-th target, with $1 \leq k < K$. Thus, given n measurements, $\bigcup_{i=0}^{K} \tau_i = \{1, \ldots, n\}$.

To describe how data association fits into the picture, we must alter our previous notation a bit to emphasize the scan-based collection of the data. Let $\mathbf{x}_{s_i}^{(k)}$ represent the state of the k-th target at scan i. The set of all state variables during the lifetime of the target is denoted by $\mathbf{X}^{(k)}$. Let the set of measurements generated during scan i be denoted by $\mathbf{Y}_{s_i}$, so that given n measurements collected after t scans, $\bigcup_{i=1}^{t} \mathbf{Y}_{s_i} = \mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$. The set of measurements that originated from the k-th target are denoted by $\mathbf{Y}_{\tau_k}$. Each target is assumed to generate at most one measurement per scan and each measurement is assumed to originate from a single target. This implies that $|\mathbf{Y}_{\tau_k} \cap \mathbf{Y}_{s_i}| \leq 1$ for $k = 1, \ldots, K$ and $\tau_i \cap \tau_j = \emptyset$ for $i \neq j$. For a treatment of split and merged measurements, see [120, 99, 64].

In order to significantly simplify multi-target models, individual targets are

[1] This form is merely a notational convenience. All of the information contained in the mapping can readily be encoded in a random variable of fixed dimension.

assumed to move independently and to be independent of any false alarms. Thus, the evolution of each target's state over time can be modeled separately. For the k-th target, that can be written as:

$$p\Big(\mathbf{X}^{(k)} \mid \mathbf{Y},\, \omega\Big) = p\Big(\mathbf{X}^{(k)} \mid \mathbf{Y}_{\tau_k}\Big)\,, \tag{1.5}$$

where the conditioning on ω tells us the which measurements in $\mathbf{Y}$ correspond to the target and thus which will contribute to the state estimate. Given those correspondences, we can then derive the model from the joint distribution, $p(\boldsymbol{X}^{(k)},\, \boldsymbol{Y}_{\tau_k})$, which is defined analogous to (1.1).

Note that in our earlier discussion of state estimation, the correspondence between the state variables and measurements was tacitly assumed to be one-to-one for the sake of simplicity, but that will not necessarily be the case. When a target is not detected during a scan within its lifetime, there may be state variables for which there is no corresponding measurement. In this case, the model is adapted by simply omitting the emission distribution for these scans. There may also be state variables for times that fall between scans when there is no possibility of having a corresponding measurement; it depends on how the model was defined. In that case, additional transition distributions are inserted without any corresponding emission distributions.

Of course, defining such a model for the evolution of a target's state first requires that the appearance of the target be identified because without this information we cannot initiate the model. And this brings up an important point to reiterate: state estimation itself does not identify targets; it only describes the time-varying behavior of targets that have already been identified. It is the role of data association to identify the targets. Depending on the algorithm, the appearance and disappearance of targets may be inferred from a global model for the collection of tracks or determined using some kind of explicit track management logic. The latter is more common owing to the ease with which it can be integrated into a sequential state estimation framework. The track management logic is typically based on likelihood ratio tests of the true target and false alarm hypotheses that are applied to sequences of measurements. For a discussion of likelihood ratio development and track scores, see [13].

Having established the presence of target k, a model based on (1.5) can be used to estimate its state given a point estimate of ω. One approach to solving the data association problem is to derive such an estimate. However, in the context of state

estimation, the mapping of measurements to their sources need not be explicit. Instead, ω can be treated as a latent parameter, described by its own probability distribution, to be marginalized out by summing over all possible values. Using this approach, the state of the k-th target can be written as:

$$p\Big(\mathbf{X}^{(k)} \mid \mathbf{Y}\Big) = \sum_{\omega} p\Big(\mathbf{X}^{(k)} \mid \mathbf{Y},\, \omega\Big)\, p(\omega \mid \mathbf{Y})\,, \tag{1.6}$$

which is the optimal Bayesian solution. The solution to the data association problem in this case requires a characterization of the distribution of the mappings, given the measurements, $p(\omega \mid \mathbf{Y})$.

Unfortunately, as the number of measurements increases, the number of possible mappings grows exponentially. Thus, there is a potential combinatorial explosion in the complexity of optimal solutions using algorithms based on either (1.5) or (1.6). This problem is most acute for high false alarm environments. The complexity is intrinsic to the nonlinear assignment problem that data association attempts to solve. Nonlinear assignment is one of the hardest combinatorial optimization problems to solve exactly [140]. In fact, it has been shown to be NP-hard [146].

One way to reduce the necessary computation is to process the scans sequentially and assume that the association of past measurements to their sources remains fixed so that only the mapping of the measurements in the current scan needs to be computed at each step. This is referred to as single-scan data association. However, the reduced computation comes at the expense of poor performance in the face of missed detections because sequential approaches make an irrevocable association decision after each scan. This means that it is not possible to revise previous association decisions in light of new information. The alternative to this approach is referred to as "deferred logic" or multiple-scan data association, which uses both current and past measurements when making data association decisions.

Single-scan data association is performed in conjunction with sequential state estimation, which yields the marginal posterior for the state of the k-th target at each time step. Depending on the approach, that can be written for scan i as:

$$p\Big(\mathbf{x}_{s_i}^{(k)} \mid \mathbf{Y}_{s_1},\, \ldots,\, \mathbf{Y}_{s_i},\, \omega\Big) \tag{1.7}$$

or

$$p\Big(\mathbf{x}_{s_i}^{(k)} \mid \mathbf{Y}_{s_1},\, \ldots,\, \mathbf{Y}_{s_i}\Big) = \sum_{\omega} p\Big(\mathbf{x}_{s_i}^{(k)} \mid \mathbf{Y}_{s_1},\, \ldots,\, \mathbf{Y}_{s_i},\, \omega\Big)\, p(\omega \mid \mathbf{Y}_{s_1},\, \ldots,\, \mathbf{Y}_{s_i})\,. \tag{1.8}$$

The second form can be rewritten to reflect the fact that only the mappings that identify the origin of the measurements in the current scan are relevant. It then becomes [135]:

$$p\Big(\mathbf{x}_{s_i}^{(k)} \mid \mathbf{Y}_{s_1}, \ldots, \mathbf{Y}_{s_i}\Big) = \sum_{j=0}^{|\mathbf{Y}_{s_i}|} \beta_{jk} p\Big(\mathbf{x}_{s_i}^{(k)} \mid \mathbf{Y}_{s_1}, \ldots, \mathbf{Y}_{s_i}, \omega_{s_i(j),k}\Big)\,, \tag{1.9}$$

where β_{jk} is the probability of the association event $\omega_{s_i(j),k}$, which describes the association of the j-th measurement in scan i with target k; $\omega_{s_i(0),k}$ denotes the event that no measurement is associated with the target. The association probability is given by:

$$\beta_{jk} = \sum_{\omega:(j\to k)\in\omega} p(\omega \mid \mathbf{Y}_{s_1}, \ldots, \mathbf{Y}_{s_i})\,, \tag{1.10}$$

which is the sum of the probability of all mappings that associate target k with the j-th measurement in scan i or no measurement when $j = 0$. Despite having been reduced to a weighted combination of single target state estimates with a single measurement, this approach to single-scan data association remains NP-hard because of the complexity required to compute the exact association probability.

At the heart of the data association problem is the posterior distribution of the mappings: $p(\omega \mid \mathbf{Y})$. The maximum a posteriori (MAP) estimate is the optimal point estimate of ω for use with models based on (1.5) and the distribution itself is required for models based on (1.11). To derive this distribution, we once again make use of the joint model for the target states and measurements, but here we treat the target states as latent parameters to be marginalized out. This can be written as:

$$\begin{aligned} p(\omega \mid \mathbf{Y}) &\propto p(\mathbf{Y} \mid \omega)\, p(\omega) \\ &= \int_{\mathbf{X}^{(1)}\ldots\mathbf{X}^{(K)}} p\Big(\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(K)}, \mathbf{Y} \mid \omega\Big)\, p(\omega) \\ &= p(\mathbf{Y}_{\tau_0}) \left[\prod_{k=1}^{K} \int_{\mathbf{X}^{(k)}} p\Big(\mathbf{X}^{(k)}, \mathbf{Y}_{\tau_k}\Big)\right] p(\omega)\,. \end{aligned} \tag{1.11}$$

The conditioning on ω in the likelihood enables us to initiate the model because it provides the number of targets and the necessary relationships between the state variables and the measurements. It is independence assumptions made earlier that justify the product factorization in (1.11). Each of the joint distributions inside of the integral can be defined analogous to (1.1). The feasibility of a closed-form solution to the integrals over the state space depends on the form of the state transition and emission distributions. For the important class of linear-Gaussian models, a closed-form solution is

indeed possible. Note that here, no explicit track management logic is necessary because the model is a global model that describes collections of any number of tracks.

Having outlined the probabilistic underpinnings of the data association problem, let us now review the major families of data association algorithms. As all practical data association algorithms are necessarily nonoptimal, the different approaches reflect different tradeoffs with regard to accuracy and computational complexity.

As alluded to earlier, data association can be viewed as a nonlinear assignment problem with constraints defined by the feasible associations. Such problems are part of an important class of NP-hard combinatorial optimization problems, which were originally studied in the context of economics, so there are many techniques available for finding solutions. Consequently, assignment-based approaches comprise a large family of data association algorithms. Since the resulting set of assignments represents a point estimate of ω, these algorithms are used with models based on (1.5). In the single-scan case, all feasible measurement-to-track associations are considered and the most likely mapping is generated. This is usually referred to as the global nearest neighbors (GNN) [13] approach because the output reflects the minimum sum of statistical distances from each track to its associated measurement. The optimization is typically performed using either the Munkres (also known as Kuhn-Munkres or the Hungarian algorithm) [128, 18] or auction [9] algorithms. In the multiple-scan case, the assignment problem becomes multidimensional, with the number of dimensions is equal to the number of scans being processed [146, 148]. Data association algorithms of this type have been developed based on a variety of different techniques including: K-best assignments [129], integer programming [127], Lagrangian relaxation [148, 147], and genetic algorithms [8]. Track management may or may not be integrated, depending on the algorithm. These algorithms tend to break down in the face of dense measurements.

Another important family of algorithms are those based on probabilistic data association (PDA). These algorithms generate approximate solutions to the optimal Bayesian single-target state estimation problem stated in (1.6). The original PDA algorithm [175] was a single-scan approach to tracking a single target in clutter. At each time step, the state estimate is computed based on a weighted combination of estimates, in the manner of (1.8). Since all measurements that are close enough to the predicted track contribute to the state estimate, PDA-type algorithms are usually referred to as all-neighbors approaches, in contrast to the aforementioned nearest-neighbor approach.

Joint probabilistic data association (JPDA) [58] is the multi-target generalization of the PDA algorithm. However, in its original form, even the JPDA approximation is computationally demanding. Not surprisingly, there are numerous lower cost variants of JPDA that differ primarily in how they compute the association probabilities and the heuristics used to enumerate the feasible associations. These include: cheap-JPDA [55], fast-JPDA [54], suboptimal-JPDA [166], and near-optimal-JPDA [164]. There is even a multiple-scan extension [165]. Track coalescence for closely spaced targets can be a problem with these algorithms and is usually addressed by intelligently pruning the association hypotheses that lead to coalescence [17, 55]. IMMPDAF [5] and IPDA [130] are the most common methods for track management for PDA-based algorithms.

One of the most popular families of data association algorithms are those based on the multiple hypothesis tracker (MHT) [160]. This popularity is due to the fact that the MHT was one of the earliest multiple-scan data association algorithms, which enables it to achieve relatively good performance in challenging environments such as those with closely spaced targets and frequent missed detections, and included a complete track management solution. The algorithm works by building and then maintaining a tree of the possible data association hypotheses. At each time step, the hypothesis with the highest posterior probability is returned. The hypotheses represent point estimates of ω which are then converted into state estimates using a model based on (1.5). In its pure form, MHT uses explicit enumeration of the hypotheses to find the best mapping. Since the number of hypotheses can grow exponentially over time, practical implementations must employ various heuristics to control this growth, such as the clustering of similar hypotheses and pruning those with low probability. Of course, all this comes at the expense of optimality since there is always the risk of eliminating the correct mapping. The major variants of the original MHT algorithm all seek to reduce its computational complexity. One version incorporates the K-best assignment algorithm [129] to maintain a fixed set of the K best hypotheses at each time step [37]. The original algorithm is sometimes referred to as hypothesis-oriented MHT. An alternative track-oriented version [109] was introduced that maintains sets of potential tracks instead of working directly with data association hypotheses, which is more efficient. Each subset of tracks corresponds to a different data association hypothesis. Structured branching MHT (SB-MHT) [40] is another variant designed to achieve computational savings in high false alarm environments. It uses a simple binary

sequential hypothesis test to determine whether a target is present in order to avoid generating too many unlikely hypotheses.

Algorithms based on the probabilistic multiple hypothesis tracker (PMHT) [180, 181, 38] treat the mapping of measurements to their sources as missing data and use the expectation-maximization (EM) algorithm [41] to a get maximum a posteriori (MAP) estimate of the state in the current scan. The EM algorithm provides an iterative solution to:

$$\arg\max_{\mathbf{X}^{(k)}} \sum_{\omega} p\left(\mathbf{X}^{(k)}, \omega \mid \mathbf{Y}\right),$$

which can been seen as a variant of (1.6). Although PMHT was not the first EM-based tracking algorithm [38] it has become the most widely used, in large part due to its computational efficiency. PMHT achieves this efficiency by making the assumption that the measurement-to-track assignments are independent random variables. This assumption is arguably incorrect, but it avoids the enumeration of possible associations and the resulting complexity that is linear in the number of targets and measurements. The original algorithm did not account for clutter, could not handle missing detections, and assumed that the number of targets was known. Over the years, numerous variants have emerged to address these limitations and more. PMHT has been extended to work in cluttered environments [196]. Turbo-PMHT [168] was introduced to speed convergence and a multiple-scan version was developed to deal with missed detections [16]. A complete track management system has also been introduced [119].

Viterbi data association (VDA) [156, 153] is the basis of yet another family of algorithms. The idea behind VDA is to use the Viterbi dynamic programming algorithm [57] to estimate the most likely sequence of measurement-to-target mappings. The process can be visualized as the search for the optimal (minimum cost) path through a trellis in which the nodes represent the measurements and the arcs represent the feasible associations. The original single-target VDA seeks a solution to:

$$\arg\max_{\omega^{(k)}} p\left(\mathbf{Y}, \omega^{(k)} \mid \mathbf{X}^{(k)}\right),$$

where $\omega^{(k)}$ denotes the subset of ω containing the mappings of measurements to the k-th target. The algorithm works by recursively computing the cost of the optimal mapping at each time step based on the results from the previous step. When the target (or processing) is terminated, the optimal path (i.e. the estimate of $\omega^{(k)}$) is recovered by

backtracking through the trellis from the final step. The underlying model is unlike those introduced earlier as it is based on the joint likelihood of the measurements and mappings conditional on the true target state sequence. In practice, the true target state sequence, $\mathbf{X}^{(k)}$, is unknown and must be replaced in the cost calculations by estimates corresponding to the path under consideration. Thus, the objective function being optimized is an approximation which can impact the performance of the algorithm. A multi-target extension known as MT-VDA [154] has also been developed. Track management is not incorporated into these algorithms.

Finally, there are the data association algorithms based on Monte Carlo methods. The earliest of these algorithms assumed that simple approximations to the association probability were sufficient and instead sought to address the problem that, in general, the state distribution given by (1.9) is multimodal and poorly modeled using a single Gaussian, which is the most common representation due to the fact that it leads to efficient state estimation algorithms. They employed the Monte Carlo method of sequential importance sampling (i.e. particle filtering) to represent the arbitrary state distribution using a set of weighted sample elements [4, 70, 96]. This approach also made it possible to use nonlinear state transition and non-Gaussian noise models. Subsequent data association algorithms focused on the association probabilities and used Monte Carlo methods to stochastically explore the combinatorial space of mappings, $p(\omega \mid \mathbf{Y})$. The resulting exploration yields a sample drawn from that space which can then be used to estimate various quantities of interest.

Markov chain Monte Carlo (MCMC) was the first such method used for this purpose and was shown to be effective for a multiple-target multiple-sensor surveillance problem [141]. MCMC is a general method for generating a sample from an arbitrary probability distribution, π, by constructing a Markov chain whose stationary distribution is π. In this context, the state space of the chain is the space of possible mappings and the sample elements are taken from the states visited. MCMC was used to estimate the association probabilities within a sequential importance sampling framework designed to propagate non-Gaussian target states [87]. It was also used as an efficient means of approximating the association probabilities in JPDA [35]. It has been shown that when the number of targets is fixed, MCMC data association (MCMC-DA) provides a polynomial-time approximation for JPDA [135]. These algorithms work with models based on (1.8). MCMC-DA has also been generalized to multiple-scan processing

using models based on (1.11). These represent a direct solution to the data association problem and require no external track management logic. The seminal algorithm was shown to converge to the full Bayesian solution and outperform MHT in challenging environments such as frequent missed detections and dense clutter [135]. Data-driven sampling can be utilized to improve the efficiency by reducing the number of samples needed to effectively explore the space [199, 201]. How best to define the required prior distribution of mappings, $p(\omega)$, when the number of targets is unknown is an open question. Nonparametric Bayesian methods have been shown to be a promising approach [60, 59].

Sequential importance sampling has also been used to draw from the posterior distribution of mappings. Algorithms of this type are based on the marginal mapping of each measurement [170, 171]:

$$p(\omega_i \mid \mathbf{y}_1, \ldots, \mathbf{y}_i)\,,$$

where ω_i denotes the indicator variable that maps measurement i to its source. Although they process the measurements one at a time in order to simplify the model, they are effectively single-scan data association algorithms because older mappings cannot be changed. A set of weighted sample elements is used to represent the distribution and each element corresponds to a particular sequence of associations. Here, the integration over the state variables needed to get the marginal mapping was motivated by the Rao-Blackwell theorem. The idea being that both the states and mapping are unknown so the general tracking problem can be posed as a joint estimation problem: $p(\mathbf{X}, \omega \mid \mathbf{Y}) = p(\mathbf{X} \mid \omega, \mathbf{Y})\, p(\omega \mid \mathbf{Y})$. By marginalizing out the state variables, the second term can be evaluated by Monte Carlo sampling and the first term evaluated analytically. According to the Rao-Blackwell theorem, the result will be a lower variance estimate [27, 46]. Although the multiple-scan MCMC-DA algorithms were derived based on different considerations, the benefits apply there as well. Recent work has extended this approach to include the simultaneous estimation of a parameter vector for the state transition and measurement models [104].

1.2.3 State Estimation Without Data Association

There are also approaches to state estimation in the general case that do not require data association as described in the previous section. Such algorithms represent a

fundamentally different way of a addressing the problem given multiple targets, missing detections, or false alarms. In situations where the sensor responses are the result of overlapping signals or when the signals from individual targets are too weak to exceed the detection threshold, these are the only approaches that can be used.

The so-called unified tracking approach generalizes the single-target filtering distribution, equations (1.2)-(1.4), to handle an unknown number of targets [179]. This is accomplished by defining the state space to be the product of $\bar{K}$ individual target state spaces, each of which consists of a continuous part and single discrete element, $\emptyset$, that indicates that the target is not in the surveillance region. The value $\bar{K}$ represents a upper bound on the number of targets and can be chosen to be any suitably large number. In this framework, the state transition distribution, $p(\mathbf{x}_i \mid \mathbf{x}_{i-1})$, accounts for both the physical motion of the targets and their appearance or disappearance because the multiple-target state encodes the information about the presence of targets. The need to explicitly map measurements to their sources is obviated by working directly with the multiple-target state. Updates in this space effectively integrate information from all possible mappings of measurements to targets [179, section 5.2.1]. Although unified tracking provides a comprehensive framework for addressing state estimation in the general case, it is not computationally tractable for most practical applications. The difficultly of computing the required products and integrals in high-dimensional spaces largely limits this approach to theoretical interest as a basis for comparison with other algorithms.

Another approach to generalizing the single-target filtering distribution is the probability hypothesis density (PHD) filter [124]. Instead of essentially stacking individual states into a single large vector as was done in the unified tracking approach, a random finite set (RFS) [122, 72] is used to represent the multiple-target state as a single meta-target. Similarly, the measurements collected by one or more sensors can be represented using a RFS and treated as the output of single meta-sensor. Using the differential and integral calculus of finite set statistics (FISST) [69], the multiple-target multiple-sensor state estimation problem can be cast in the familiar single-target single-sensor filtering framework. Unfortunately, the FISST filtering recursion requires the evaluation of combinatorial sums of high-dimensional integrals and is generally computationally intractable. In practice, the exact recursion is approximated by one that propagates the PHD of the multiple-target state [124]. The PHD is the first-order sta-

tistical moment of the RFS, which is equivalent to its intensity. Thus, the integral of the PHD for the multiple-target state yields the expected number of targets in the region of integration and the peaks in the PHD reveal the most probable values of individual target states. However, even the propagation of the state PHD presents computational challenges because the recursion still consists of multiple integrals that have no closed-form solutions in general.

The first practical implementations of PHD filters were based on sequential Monte Carlo and used a large set of weighted particles to represent the state PHD [189, 177]. Clustering of the particles is then performed to extract the state estimates. Later, a more efficient, but more limited, implementation was developed by modeling the state PHD as a Gaussian mixture which admits closed-form evaluation of the filtering recursion [188, 187]. This eliminated the problem of having to identify the peaks in the PHD since they are given by the means of the mixture components and made extracting state estimates more reliable. However, despite its efficiency compared to the particle-based variant, algorithms based on the Gaussian mixture PHD also suffer from computational problems as the number of mixture components grows rapidly and without bound. Practical implementations must employ strategies to control this growth through merging and pruning as is done in MHT data association. In the presence of missed detections and high false alarm densities, PHD filters were found to produce unstable estimates of the number of targets (i.e. the cardinality of the underlying RFS). To address this problem the cardinalized PHD filter [121, 123] was introduced which jointly propagates the distribution of the number of targets along with the state PHD. A cardinalized Gaussian mixture PHD has also been developed [190].

While PHD filters offer the advantage of sidestepping the NP-hard data association problem by directly propagating an approximation to the multiple-target state, this is not without a cost. Since they do not record target identities, PHD filters provide no means of associating the individual state estimates between scans and hence cannot produce tracks. This is not desirable for applications in which it is necessary to identify the trajectories of individual targets. Thus, researchers have studied ways combining PHD filtering with data association [116, 138, 32, 139]. In practice, this involves associating the state estimates, derived from the particle clusters or the Gaussian mixture components depending on the implementation, with target labels using methods analogous to those discussed in section 1.2.2. Managing a tree of hypotheses in the manner of

MHT is the most common approach. Compared to the actual MHT algorithm, however, the computational demands are much lower because the set of state estimates is typically smaller than the number of measurements and no additional filtering is necessary. Recent work has proposed a track-oriented variant of PHD that uses RFSs to represent multiple-target trajectories rather than multiple-target states [182].

1.3 Related Work

The focus of the present research is twofold. First, we seek to further develop the concept of tracking as a data association problem, with a particular interest in modeling the relevant uncertainty. Second, we seek to address a specific, and increasingly important, problem in wide-area persistent surveillance applications: tracking people using radar. The related work in the literature will be considered separately for each of these areas. However, let us first put forth some of our assumptions and discuss some of the special challenges presented by the motivating application in order to concentrate on the appropriate prior art.

1.3.1 Assumptions

The targets of interest are assumed to be individual human walkers moving in an outdoor non-urban environment. Tracking people in urban environments, especially using radar, presents unique challenges (e.g. highly constrained line of sight visibility and multi-path reflections) that will not be considered here. While recognizing that humans are by no means limited to walking, dynamic models for running or foot-speeds beyond that of a typical walker will not be considered here either. Targets are assumed to move independently; group dynamics and models of target interaction will be left for future work.

Most approaches to people tracking assume that some appearance information is available from which measurements can be classified or associated. However, in the present work humans are treated as "small targets" [48] or "point targets" as they are commonly known. Point targets are defined as those whose measurements do not include any information about the target's appearance, features, or attributes. That is to say, targets for which we only have kinematic measurements, position and its derivatives, and possibly information related to the confidence of the detection (e.g. signal strength).

Such targets are essentially indistinguishable given a single measurement. Only by observing specific time-varying patterns in the data that correspond to the target's motion, can they be recognized. Here, we further restrict our attention to point targets for which only position measurements are available. This is due to the fact that this research was originally inspired by a surveillance application that employed low cost non-coherent radars, which cannot exploit the Doppler effect to provide (radial) velocity information. However, since position is the lowest common denominator, the point target assumption establishes a performance baseline that is applicable to any detector. Future studies can then investigate the potential gains from additional measured values.

The surveillance region is assumed to be periodically scanned using a single radar. Sensor scheduling and the fusion of information from multiple sensors or different sensor modalities will not be considered here. The radar is assumed to be tower-mounted or airborne to achieve sufficient coverage. For mobile airborne radar, the platform motion is assumed to be relevant only to the detector. Regarding the radar itself, details such as the type of radar, operating frequency, or antenna size and type are assumed to have been appropriately chosen for the detector and the given environment. The only constraint is that a single target is assumed to generate one measurement. That is to say, there are assumed to be no split measurements. Due to the resolution of a typical radar system, closely spaced targets are, however, expected to result in merged measurements.

1.3.2 Challenges

Most tracking algorithms were initially developed for air targets. Tracking ground targets, on the other hand, presents additional challenges [31], especially when the targets of interest are people.

The first challenge is a spatially varying probability of detection (P_D) that leads to sparse measurements. In order to detect a target, its return must be separated from the clutter returns which will vary in strength and number depending on the background environment. Thus, the reliability of the detection process is spatially dependent. Where the radar signal is blocked by terrain features, vegetation, or man-made structures, detection becomes impossible. This effectively reduces the P_D there to zero for any threshold above the noise floor. Furthermore, the area of these shadowed regions can be significant even given an optimal placement of the radar. It is also

likely that some targets will take advantage of the visibility restrictions in an attempt to evade detection. Unlike vehicles, people are small enough and maneuverable enough to effectively exploit this advantage when possible. Even in the absence of any active evasion, people are more likely to be naturally obscured by the terrain and vegetation given their size. A consequence of this limited visibility is that the tracker may have to endure long periods without any measurement updates when a target enters one of these blind spots.

Unfortunately, even in areas with full sensor coverage, achieving a sufficient P_D with an acceptable probability of false alarm (P_{FA}) can be difficult for human targets. Compared to vehicles, people have a relatively small radar cross section (RCS) and they effectively change shape constantly as they walk with the movement of the head, limbs, and torso. They reflect only a small amount of energy back to the detector, primarily from the torso, and their non-rigid body motion causes the radar echo characteristics to change with their stride. People are also relatively slow and prone to stopping, which makes them difficult targets to discriminate against background clutter. The standard approach, and the basis of the Ground Moving Target Indicator (GMTI) mode, of using the Doppler velocity to distinguish moving targets from stationary clutter becomes less reliable as their in-range speed goes to zero [43]. With this approach, there is a trade-off between minimizing clutter returns and the minimum detection speed of the target [21, 162]. However, one of the ways a search radar can improve the P_D (for a given P_{FA}) is to reduce the scan rate in order to increase the dwell time to put more energy on the target. Here, there is a tradeoff to be made between P_D and the revisit rate. If we favor an increased probability of detection to compensate for the aforementioned detection difficulties, the tracker must again endure sparse measurements.

The next challenge is a high false alarm rate. Unlike the sky, the ground is rich with potential sources of unwanted radar returns. This so-called clutter complicates the detection of ground targets and is a major source of false alarms. Wind blown foliage, for example, can produce a Doppler spectrum much like a typical human walker.

Limited visibility not only impacts the sparsity of the measurements, but it also exacerbates the problem of false alarms because it reduces the overall expected probability of detecting the target in a single-scan. Thus, if we do not want to give up too much P_D in the visible regions, then we need a tracker that can handle potentially large numbers of false alarms. The option of boosting P_D using a detection threshold

that increases the false alarm rate of the detector is only viable if the tracker can mitigate the false alarms and avoid wasted resources.

Finally, the last challenge is maneuverability. The motion of even a nimble fighter plane is relatively constrained by aerodynamics. By comparison, ground targets are considered to be highly maneuverable; people even more so than vehicles. Within a typical detection period of 30 seconds to 3 minutes (as observed in sample data), a person can start, stop, change their speed to any fraction of their maximum or change their heading to any direction. Said another way, the maneuver time constant is much less than the sample period. The environment might restrict some of these possibilities, but in general, human motion is relatively kinematically unconstrained when observed on this time scale. Most tracking applications discussed in the literature assume frequent measurement updates so they are able to use dynamics models based on the target kinematics that are reasonably accurate over short intervals. A small amount of random noise is enough in this case to account for modeling errors. Here, we do not have that luxury.

1.3.3 Tracking as Data Association

Given the intimate relationship between state estimation and data association, there are many tracking algorithms that might be said to focus on the data association problem. We can broadly divide these algorithms into two categories based on the type of output they produce. Those in the first category only produce a point estimate of ω representing a single set of data association hypotheses that conveys no information regarding the uncertainty. An example of this would be a correlation tracker designed to associate detections of the same object based on their similarity in appearance. There are numerous such algorithms in the literature with the stated objective of determining these associations. However, the focus on data association here is essentially semantic as the results are no different than what we would obtain if the nominal emphasis was on state estimation. Typically, this difference in emphasis follows from the fact that the measurements themselves describe the most important aspects of a target's state, such as its position, with sufficient precision. Of course, even in this case the associated measurements are generally still subject to state estimation in order to infer unmeasured components of the state, such as target velocity.

Algorithms in the second category produce a distributional estimate of ω based

on a statistical model of the data association problem itself. Since this cannot be done in closed-form, such estimates will consist of a sample from the modeled distribution from which we can extract the desired information. These algorithms represent the essential prior art for the current research. There are a number of precedents for performing inference on the posterior distribution of mappings, and we will remark on where and why the associative tracking model proposed in the next chapter deviates from them.

Pasula et al. [141] appear to have been the first to use a sample from $p(\omega \mid \mathbf{Y})$ to solve the data association problem in a tracking application. In the context of multiple-sensor traffic surveillance, they showed that a sample from the posterior distribution of the mappings generated using Markov chain Monte Carlo (MCMC) can be used to approximate the expected value of some function, $f(\omega)$, with respect to $p(\omega \mid \mathbf{Y})$. Their algorithm essentially addressed multiple-scan data association using a model in the form of (1.11). Here, the latent states were introduced into the model for the express purpose of rendering the observations from different cameras conditionally independent, rather than being an integral part of the concept and something that we might want to recover given a point estimate of ω, but the end result is analogous. However, they deviated from the usual assumption that the transition distribution (including its initial value) governing the evolution of the latent variables is known and unchanging, and re-learn those parameters whenever a new target is observed. The parameter learning is performed using expectation maximization (EM) [41] (in a process known as the Baum-Welch algorithm) where the E-step employs MCMC to generate a sample from $p(\omega \mid \mathbf{Y})$ that is then used to compute $E[\mathbf{X} \mid \omega, \mathbf{Y}]$.

All targets (i.e. vehicles) were assumed to have been detected at each opportunity and generate one and only one measurement per sensor. There were assumed to be no false alarm measurements. Thus, determining the appearance and disappearance of targets was not an issue in this application. Instead, the focus was entirely on the so-called object identification problem, which is equivalent to determining the origin of each measurement and hence equivalent to the data association problem. The MCMC sampler was implemented using the Metropolis-Hastings algorithm (see section 2.4.1) with a simple proposal based on swapping assignment pairs across one pair of sensors. Experimental evaluation was performed using three sensors and approximately one hundred measurements. They concluded that the sampler chain converged "very quickly" although no convergence assessment was provided. The work of Pasula et al. was an

important first step in validating the concept of MCMC data association (MCMC-DA) but leaves open the critical question of how to deal with false alarms, which makes determining the appearance and disappearance of targets non-trivial.

Oh et al. [135] provided the first comprehensive solution to MCMC-DA that could handle missing measurements, false alarms, and track management. They proposed both single-scan and multiple-scan algorithms. The single-scan algorithm, however, assumed that the number of targets was fixed and known, perhaps based on some external track management logic. It used MCMC and the Metropolis-Hastings algorithm to generate a sample from:

$$\begin{aligned} p(\omega \mid \mathbf{Y}_{s_1}, \ldots, \mathbf{Y}_{s_i}) &\propto p(\mathbf{Y}_{s_i} \mid \mathbf{Y}_{s_1}, \ldots, \mathbf{Y}_{s_{i-1}}, \omega)\, p(\omega \mid \mathbf{Y}_{s_1}, \ldots, \mathbf{Y}_{s_{i-1}}) \\ &= p(\mathbf{Y}_{s_i} \mid \mathbf{Y}_{s_1}, \ldots, \mathbf{Y}_{s_{i-1}}, \omega)\, p(\omega)\,, \end{aligned}$$

at each scan. The likelihood term is given by:

$$\begin{aligned} p(\mathbf{Y}_{s_i} \mid \mathbf{Y}_{s_1}, \ldots, \mathbf{Y}_{s_{i-1}}, \omega) = \prod_{k=1}^{K} \int p\Big(\mathbf{Y}_{\tau_k(s_i)} \mid \mathbf{x}_{s_i}^{(k)}\Big) \\ \left[\int p\Big(\mathbf{x}_{s_i}^{(k)} \mid \mathbf{x}_{s_{i-1}}^{(k)}\Big)\, p\Big(\mathbf{x}_{s_{i-1}}^{(k)} \mid \mathbf{Y}_{s_1}, \ldots, \mathbf{Y}_{s_{i-1}}, \omega\Big)\, d\mathbf{x}_{s_{i-1}}^{(k)}\right] d\mathbf{x}_{s_i}^{(k)}, \quad (1.12) \end{aligned}$$

where $\mathbf{x}_{s_i}^{(k)}$ denotes the kinematic state of the k-th target during scan i and $\mathbf{Y}_{\tau_k(s_i)}$ denotes the corresponding measurement based on the mapping ω. Note that only the mappings of the measurements in scan i change during sampling; the rest of ω remains fixed. The prior distribution of the mappings, $p(\omega)$, was defined based on a Poisson false alarm model and the assumption of a spatially uniform probability of detection (P_D). The proposal distribution maintained the status quo with a probability of 0.5, otherwise a pair of measurements was selected from which an association was added, deleted, or switched, depending on their initial mappings. The pair of measurements was selected uniformly at random with one measurement chosen from among those recently assigned to targets and the other chosen from among those in the current scan and validated for the given target. Measurement validation was used to determine which measurements could plausibly be associated with a given target in order to reduce computation. For target k the determination was made by applying a fixed threshold to (1.12). Bounds on the rate of convergence were proven and it was shown that the algorithm provides a polynomial-time approximation to JPDA.

The multiple-scan algorithm relaxed the assumptions of a fixed and known number of targets and thus provided a complete solution to the general data association problem. It used MCMC and the Metropolis-Hastings algorithm to generate a sample from the posterior distribution of the mappings based on a model in the form of (1.11). The prior distribution of the mappings, $p(\omega)$, was again defined based on a Poisson false alarm model and the assumption of a spatially uniform P_D. However, here it included two additional terms based on a Poisson target appearance (birth) model and the probability of a target disappearing in a given scan. The proposal distribution consisted of a mixture of five reversible moves (or pairs of moves for those that are not self-reversible): add/delete, split/merge, extend/reduce, update, and switch. To improve the performance of the algorithm, the moves made use of a neighborhood tree defined around each measurement. The neighborhood tree effectively represented the feasible associations as determined by a distance-based gating criteria, where the maximum distance was computed by considering the maximum speed of any target and the number of consecutive missing measurements that would be tolerated in a track.

Having provided a complete probabilistic solution to the data association problem for multiple-target tracking, the work of Oh et al. had a significant influence on the present research. The associative tracking model, which will be presented in the next chapter, builds on their multiple-scan approach to address its limitations with respect to our motivating application. There are two changes to the basic model. The first is to accommodate a spatially varying P_D and the second is to the prior distribution of the mappings. Their prior concentrates a lot probability around the expected number of targets, which is given by the birth rate parameter. This is poorly suited to surveillance applications because the number of targets is unknown and one of the very things that we wish to learn from the model. In practice, we also found it difficult to determine an objective value for the probability of disappearance. This was complicated by the fact that the overall performance did not always vary predictably with changes in these values. How best to represent our ignorance regarding the possible mappings is an open question for which we do not have a specific answer. Thus, we elected to study this aspect of the problem in detail in chapter 3 and then evaluated a number of promising candidates.

Beyond the changes to the model, we also sought to improve the inference algorithm. When applied to our data, the proposal distribution used by Oh et al. of-

ten suffered from a very low (under 5%) acceptance rate. The sampler also exhibited significant sensitivity to the initial state, which raised concerns about about how efficiently it was exploring the space of mappings and whether our posterior estimates would converge given a realistic sample size. They indicated that a greedy initialization scheme was used that may have started the sampler in a high probability initial state and therefore masked this problem. The performance bottleneck was primarily the add move, but the extend and update moves were similarly problematic because they used the same procedure for selecting the false alarm measurements that are to become target measurements in the proposal. At issue is the probability with which the selection procedure results in a favorable sequence of measurements under the model. An additional complication is that the selection procedure itself has parameters that must be tuned. This motivated us to consider a data-driven proposal that makes use of a generalization of the neighborhood tree that we call the feasible graph. When taken together, the neighborhood trees for each measurement comprise a graph that encodes the feasible associations. Oh et al. introduced the concept of using such a structure derived from the set of measurements to improve the performance of the proposal distribution. In their case, the collection of neighborhood trees helped the sampler avoid making proposals with negligible likelihood by requiring the presence of an edge between consecutive measurements mapped to the same target. We extend this concept by first computing a weight for each edge that is related to the expected likelihood. Then we exploit the directed acyclic structure of the graph to efficiently find minimum cost paths, where the cost is an approximation of the negative posterior probability, that are likely to correspond to tracks.

Recognizing the potential problems with using a Poisson target appearance (birth) model when the number of targets is unknown, Fox et al. [60] proposed using a nonparametric prior for $p(\omega)$ based on the Dirichlet process (DP) for multiple-scan data association models. The details of this and related prior models are covered in chapter 3. For now let it suffice to say that it provides a principled probabilistic approach to modeling ignorance, with respect to the number of targets, in a way that lets the data drive the creation of tracks. They derived the model for $p(\omega \mid \mathbf{Y})$ by assuming that each measurement, conditional on a parameter representing the latent state of the underlying target, could be described by a Gaussian mixture with one component for each target. The weight of each component is the probability that the measurement originated from

the corresponding target. Since the number of targets, and hence the number of mixture components is unknown, a Dirichlet process mixture (DPM) [131] was used to model a countably infinite number of targets. A DPM places a Dirichlet process (DP) prior on the countably infinite set of mixture weights. Those weights are then integrated out in closed-form to get the distribution of the implied mappings of targets to measurements, $p(\omega)$. The parameters representing the states were assumed to be Gaussian so those too were integrated out in closed-form to get the likelihood, $p(\mathbf{Y} \mid \omega)$. Inference was based on a posterior sample generated using MCMC via the Gibbs sampling algorithm (see section 2.4.1).

Their work established two important precedents. The first was the use of nonparametric prior models for tracking applications in which the number of targets is unknown. Although the idea seems obvious, the second precedent is the use of the posterior sample for something other than generating a single collection of state estimates. In other words, Fox et al. appear to have been the first to characterize the uncertainty of some relevant aspect of the tracking problem by considering the target crossing ambiguity. However, despite these important contributions, their algorithm does not provide a complete solution to the general multiple-target tracking problem. The primary deficiency is the lack of a false alarm model, which is necessary for any practical application, and traditionally this has been something that is incorporated into the prior distribution of the mappings. In addition, the lack of a P_D model is also problematic and traditionally this too has been incorporated into the prior as was the case in Oh et al. Unfortunately, nonparametric priors, such as the DP, are not amenable to such modifications. Finally, their algorithm does not address the critical issue of how to determine a suitable value for the one and only parameter of the DP prior. The DP has a scalar concentration parameter that controls its tendency to form new clusters. The resulting prior probabilities are sensitive to the choice of this parameter so it must be chosen carefully. Since the concentration parameter affects the expected number of clusters, which of course is unknown, it should be treated as a random variable in the model with its own hyperprior. How best to do this is an open question. Our associative tracking model addresses each of these issues so that nonparametric priors can be used for data association in the general case. This is accomplished by introducing a set of false alarm indicator variables, one corresponding to each measurement, that allow us to factor the prior over mappings into a false alarm component and a target

cluster component. These indicators are defined so that they are consistent with the mappings in ω and are thus redundant in terms of information; serving only to factor the prior so that target clusters and false alarm counts can be treated separately. The treatment of the spatially varying probability of detection is incorporated into the measurement likelihood model because the P_D is now dependent on the position state and thus independent of $p(\omega)$. A principled hyperprior for concentration parameter is also proposed.

Even if we accept the model from Fox et al. as is, there is also a matter of practical concern, namely the efficiency of the sampler. They opted to use a Gibbs sampler, which has a certain appeal because the definition of the proposal distribution is well-defined and thus spares the practitioner the trouble of selecting one with the right properties. However, this often comes at the expense of slow movement through high-dimensional spaces especially when the sampler must pass through low probability regions. The Gibbs sampler is a common choice for DPM models because the prior can easily be written as the probability of one cluster assignment (i.e. mapping) conditioned on all the other assignments. However, this does not make it the optimal choice in practice. Fox et al. were aware of the potential for slow convergence because they suggested the inclusion of a track switching step to speed things up, but using our data this was not sufficient to prevent unacceptably slow convergence and extreme sensitivity to the initial state for realistic sample sizes. Hence, our decision to work with mixture proposals like those used by Oh et al. Such proposals enable large and meaningful moves to quickly explore the space of mappings.

The use of data-driven proposals to improve the efficiency of sampling from multiple-scan data association models was introduced by Yu et al. [199, 201]. However, beyond introducing that concept, their work is not germane to the present research. They addressed object tracking in video surveillance applications with measurements defined as entire foreground regions extracted by a background subtraction algorithm. Although they adopt the familiar notation for ω, which represents sets of indices mapping each measurement to its source, they have defined it to represent not only the mappings but shapes (rectangles) positioned to cover the spatial extent of each target as well. The data-driven part of their proposal distribution does not affect how the mappings are updated. It is only used to propose changes to the covering rectangles by exploiting local classifier models. Regarding the proposal of new mappings, they

do employ extend/reduce moves that can operate both forward and backward in time, which we have adopted as well.

Their results suggest that such data-driven proposals are effective and confirm the importance of using appearance information, when available, in data association models. Unfortunately, the quality of the resulting sample remains an open question because they focus exclusively on the maximum a posteriori estimate. It is not clear why a sampling algorithm is preferable to a stochastic optimization algorithm for the purpose of obtaining such a point estimate. Lastly, their model of $p(\omega)$ is a classic example of what we call property-based priors. Such priors are constructed by taking the product of functions that each express a preference for some property that we wish to see in ω, without regard for the combinatorial structure. These will be discussed in chapter 3.

Although clearly not the authors' intention, Yu et al. appear to have provided an example of a discriminative model [132] for data association. In other words, they essentially modeled $p(\omega \mid \mathbf{Y})$ directly without describing the data generating process in terms of a properly defined likelihood and prior that together model the joint distribution of the measurements and ω. It is never made precisely clear what their definition of ω looks like in terms of random variables, which only adds to the confusion. However, their description of the appearance "likelihood" component is plainly discriminative in its construction as it describes the probability of the association and coverage of the rectangles given the data. In other words, the appearance likelihood is defined using the probability that the covered image region is a target. They rely on the fact that, as a function of both ω and $\mathbf{Y}$, this can be normalized such that when integrated over the measurement space it sums to one. Their inclusion of a prior term does not change our opinion that the overall model is more appropriately termed discriminative. This, of course, is a perfectly valid approach for applications that only require inference with respect to $p(\omega \mid \mathbf{Y})$. And there are valid arguments as to why this may even be preferable. The model should just be described accurately in any case. Later, when discussing future work, we will advocate for the use of such discriminative models as a way to incorporate target intent and high-level behaviors that would be intractable using a generative model.

Ge and Collins [61] adopt a similar approach to that of Yu et al., which includes using what is appropriately termed a discriminative model. However, instead

of processing the raw detections they use trajectory fragments computed over multiple scans (e.g. 30) as the measurements. They also address the case of unknown parameters in the prior distribution of the mappings by introducing an uninformative hyperprior into the model. We too will employ such a hyperprior, which is a standard part of any Bayesian hierarchical model.

Sequential Monte Carlo has also been used to recursively generate samples from the posterior distribution of mappings. Särkkä et al. [170, 171] proposed using particle filter to perform what is effectively single-scan data association. We qualify that term because the measurements are actually processed sequentially in time to estimate the marginal posterior of each mapping. Mappings assigned during a previous scan are irrevocable and will not change in light of new information. In this sense, the algorithm behaves like a single-scan algorithm. However, the sequential processing limits the information that will influence a given mapping to those measurements within the scan that were processed earlier in time. Normally, the information conveyed by all of the measurements within the scan would be taken into account. On the other hand, the sequential formulation does provide a significant advantage in terms of reduced computational requirements and the efficiency of the sampler. These gains come from the fact that it is possible in this context to generate the sample by drawing from what is known as the optimal importance distribution, which minimizes the variance of the resulting estimate. This does, however, require one notable simplifying assumption: the disappearance of targets in restricted to one per time step. By drawing from the optimal importance distribution, there is no need to search for an effective proposal distribution as we must when using MCMC. For applications in which we are able to forego the benefits of multiple-scan processing, their algorithm provides a compelling approach.

An important contribution of their work was to offer a justification for integrating out the latent target states based on the Rao-Blackwell theorem as we noted at the end of section 1.2.2. This provides a convenient way to think about the relationship between state estimation and data association. They also foreshadowed the use of nonparametric priors by suggesting a form for the conditional distribution of the current mapping, given the previous mappings, that is equivalent to the uniform process (UP) prior that will be discussed in chapter 3.

1.3.4 Wide-Area Persistent Surveillance

The application motivating the present research can be broadly defined as wide-area surveillance of moving ground targets. This is currently an area of increasingly active research as the need for solutions grows, along with processing power and the capabilities of sensors.

Most of the work to date has focused on using electro-optical/infrared (EO/IR) or radar sensors to detect and track, vehicles and people (or "dismounts" in military parlance). Although the use of acoustic and seismic sensors has also been studied, we shall not consider these technologies here as they are generally deployed in sensor networks with an emphasis on localization accuracy and fusing information from the different sensors. Our primary interest, of course, lies in the particular combination of radar sensor and people tracking because it is both relevant and in need of further study.

Having already considered the matter of how the tracking problem itself is posed, here we shall survey the existing research with an eye toward how related ground target surveillance problems have been addressed. Of particular interest are graph-based representations of the possible associations, the target dynamic models, and the treatment of missing measurements.

1.3.4.1 EO/IR

EO/IR sensors have found widespread application in ground target surveillance. They are frequently employed for both people and vehicle tracking because of their high spatial resolution, which facilitates target classification and data association. Often, EO/IR sensors are used in conjunction with radars to augment the radar's relatively poor spatial resolution. Typically, this involves using the radar contacts to slew the EO/IR sensors to the targets.

In the present research, we have adopted a view of humans as point targets (defined earlier) which puts us at odds with most of the literature on people tracking with EO/IR sensors. Targets observed with this modality are generally detected using background subtraction techniques or an object detector trained to recognize the target's appearance. Both approaches identify a region of the image that can be analyzed based on shape, edges, texture, etc. The ability to provide this appearance information is

one of the primary virtues of EO/IR sensors. It is used extensively to determine which measurements are associated with which targets, often in conjunction with motion-based constraints. When the resolution is sufficient, appearance information is also used to estimate target pose and other attributes along with the kinematic state. In video surveillance applications, the trajectories of human targets are often defined as sequences of poses that are described by dynamic models that include joint angles and limb motion (e.g. [56, 158]). Unfortunately, none of these developments are applicable to point target tracking.

Background subtraction is the most common detection strategy in wide-area video surveillance applications. This is because the small size of humans makes it difficult to construct robust object detectors and the movement of the target relative to the background is a defining feature. But with any strategy, reliably detecting people without generating an excessive number of false alarms is a challenging problem. People may exhibit minimal contrast with the background and may cover just a few pixels, or less at some ranges. Movement in the background or variations in appearance due to lighting (or temperature) changes complicate the process. However, despite the potential difficulties that come with dense false alarms or low probability of detection, the research remains focused primarily on detection algorithms. The uncertainty with respect to the origin of the detections is generally assumed to be such that a point estimate of the associations, determined largely by the appearance information, is sufficient. A prime example of this comes from Yu et al. [201], whose work we discussed in the previous section. Despite treating tracking as a data association problem and using Monte Carlo methods to draw a sample from the posterior distribution of mappings, they sought only the maximum a posteriori estimate. Their experiments did show that the benefits from incorporating appearance information, when possible, can be significant. It should be noted that the use of Monte Carlo methods in this regard is not typical. Assignment-based data association algorithms appear to be the most common due to the ease with which appearance information can be incorporated into the cost matrix (e.g. [86, 161, 98]).

There is existing work on tracking point targets using IR sensors, which are sensitive enough to pick out humans at long range. Often, this requires that the raw signal be available so that detection and tracking can be combined into a single process, known as track-before-detect (TBD), in which the signal is integrated over time in order

to boost the signal-to-noise ratio for weak targets. Approaches to state estimation without data association, such as those discussed in section 1.2.3, are necessary in this case. However, we will not concern ourselves with that here as our interest lies in how the data association problem has been addressed given point target detections. Zaveri et al. [204] addressed it in the context of multiple point targets in a dense clutter environment by using a genetic algorithm to calculate the assignment weights. Given those weights, a JPDAF-based approach was used to derive the state estimates. In a later publication [205], those same researchers also considered a PMHT-type algorithm for the same application. Like the PMHT, their approach uses EM to approximately integrate over all possible associations. However, unlike PMHT, all validated measurements are used to update the target state instead of the centroid of measurements in order to improve the performance in a dense clutter environment. Nichtern and Rotman [133] used a dynamic programming algorithm (DPA) to perform the data association. We did not find any examples of existing algorithms that produced a sample-based estimate of the measurement to target mappings.

Although not formulated as a solution to the data association problem and designed to exploit appearance information, which we do not have, there are EO/IR tracking algorithms that overlap to some extent with aspects of our proposed approach.

Cohen and Medioni [34] considered the problem of tracking unspecified objects from a moving airborne platform. Their algorithm incrementally constructs a graph representation of the detections that is equivalent to our feasible graph (see section 2.4.3.1). The edge costs were defined to be the likelihood that the connected detections correspond to the same object. Object trajectories then correspond to paths through the graph, for which each can be assigned a cost. They assumed that objects can generate multiple detections and exploited the graph structure to cluster such detections in a way that is less likely than traditional methods to merge detections of objects with distinct trajectories. To extract the object trajectories, each graph node without a parent was considered a potential object source from which the path with the optimal cost was determined by traversing the graph. Zhang et al. [206] constructed a network flow graph from the detections in a video surveillance application and then found the optimal set of paths using a minimum cost flow algorithm. Yan et al. [197] used a graph to represent the relationships between short trajectory segments, estimated using three consecutive frames, in an algorithm for tracking tennis balls in video sequences. The edge weights

were set to reflect the compatibility of the connected segments. They first solved the all-pairs shortest path problem and then used a greedy algorithm to select the best paths from among the set of shortest paths. These algorithms demonstrate the effectiveness of using graph representations of the data in order to facilitate the efficient extraction of paths that correspond to target tracks. We will employ this technique as well, but within a Monte Carlo sampler. In our algorithm, the graph representation serves as an approximation to the global statistical model for the distribution of associations from which we are sampling. Its structure enables us to efficiently generate meaningful proposals, especially those that contain new tracks, that encourage a rapid exploration of the sample space. Those proposals are then be accepted or rejected based on their agreement with the target distribution of the sampler.

Reilly et al. [161] considered the problem of tracking large numbers of vehicles in dense traffic from a moving airborne platform. Although they did not define the vehicles as point targets, they assumed that the target resolution was insufficient to determine the associations between measurements based on appearance information forcing them to rely on velocity models. Noting that these models were not precise enough, especially given the target density, they employed road-based constraints. In chapter 4, we will similarly supplement velocity models with path-based constraints for human walkers. Although, the path-based constraint will be a soft constraint designed to express a preference for path following while still accommodating off-path motion. They addressed the data association problem using a single-scan assignment-based algorithm. Since this would be intractable given the entire set of measurements, the scene was divided into grid cells and then multiple-target tracking was performed within each cell. The tracks were then linked across cells.

Many of these algorithms rely on what are called tracklets. Broadly speaking, these are small fragments of a complete track, but there is no universally accepted definition of exactly what that means. In some cases, tracklets are the result of applying a standard tracking algorithm over a small fixed window of data (e.g. [197, 2, 86, 151]). The term has also been used to describe the fragments of larger tracks broken up by significant occlusions or sensor gaps (e.g. [98, 200, 142]). In at least one case, tracklets are defined to be tracks based on a limited number of measurements from a single sensor and whose state estimate errors are not correlated with any other data from the same target in a multi-sensor system [49]. When we describe the feasible graph

in the next chapter, we will introduce our own notion of a tracklet which is defined as a pair of measurements that originated from the same source. Thus, our tracklets are minimal track fragments. They correspond to the edges in the feasible graph and convey local information about the possible associations that will be exploited by the proposal distribution of the sampler.

Despite the variation in how they are defined, tracking algorithms that work with tracklets all ultimately seek to stitch them together into complete tracks. Our algorithm is no exception. The advantage of this approach, as noted by Kaucic et al. [98], is that it significantly reduces the space of association hypotheses; either as a result of replacing the original measurements with the tracklets or using the original measurements and truncating the space by excluding associations not covered by a tracklet. There are, of course, many different ways to proceed. Kaucic et al. and Perera et al. [142] linked tracklets that correspond to the same target using assignment-based data association with a cost matrix computed using kinematic and appearance models. Huang et al. [86] used a three-level hierarchical framework to combine tracklets into tracks. At the low level tracklets are generated based on pairwise associations. At the middle level, those are combine using an assignment-based data association algorithm. Finally, at the top level the probabilities of target appearances and disappearances, and occlusions are estimated, which is used to refine the target trajectories. Singh et al. [178] grew each tracklet using MHT data association. Our approach is similar to that of Yan et al. [197] where the tracklets are related to a graph structure and the problem of stitching them together becomes one of finding paths through the graph with the desired properties.

Kaucic et al. [98] sought to address the difficulties caused by long occlusions and sensor gaps in video surveillance applications. To aid in this purpose, their algorithm included a scene understanding module designed to identify the areas where targets are likely to become occluded. This information was then used to suspend occluded tracks (i.e. tracklets) which could then be reconnected by a track matching module. Although we will not follow such an approach ourselves, this work is relevant because it is an example of a tracking algorithm designed to address the problem of spatially varying probability of detection (P_D) that we too will be facing. Our approach is to use a measurement space augmented to represent the possibility of no detection (see sections 2.2 and 2.3.1). The likelihood is then defined based on a measurement model

and a spatially varying P_D model, similar to their occlusion map, which must be learned offline.

1.3.4.2 Radar

Currently, radar sensors are used overwhelmingly for vehicle tracking in ground target surveillance. Tracking people with this modality has been largely limited to near-range applications, including indoor surveillance where the ability of certain radars to make detections through walls is exploited.

We suspect that research into wide-area people tracking using radar is relatively limited at this point in time due to a lack of available data; at least in the open research community. When observed at a distance, human targets present a challenging detection problem that is still very much an open research topic [83]. Most existing radar systems cannot reliably detect them at long range because of their small radar cross section (RCS) and low speeds. The average human walking speed often falls near or below the minimum detectable velocity of the ground moving target indicator (GMTI) mode, which is set in order to minimize clutter returns. Depending on the capabilities of the tracker, the minimum detectable velocity may be set rather conservatively so as not to overwhelm the algorithm with false alarms.

However, humans are an important class of targets and a growing number of modern radar systems are specifically designed to detect them. These systems take advantage of improvements in electronics and, most importantly, available processing power. Such capability has become known as the dismount moving target indicator (DMTI) mode to avoid confusion with the traditional GMTI, which for all intents and purposes is only reliable for vehicles. An example of one of the earliest such radars in production is Northrop Grumman's Vehicle and Dismount Exploitation Radar (VADER), which was successfully demonstrated in 2010 [68]. As more DMTI-capable radars are fielded and data sets become available to the broader research community, we expect to see more focus on wide-area people tracking using radar.

While the details of the detection algorithm vary with the implementation, the GMTI generally implies discrimination based on the Doppler velocity. The DMTI, on the other hand, has come to imply discrimination based on the micro-Doppler signature [74, 75, 183, 30, 83, 62, 42]. Micro-Doppler refers to the features in the Doppler spectrum produced by the movement of a target other than its global translation. As humans

move, their changing shape produces a micro-Doppler signature that is distinct from those produced by other ground targets such as vehicles or animals. This signature can be identified to facilitate detection, even in cases where the target is moving cross-range and the Doppler velocity is negligible [183].

Although we adopted the view of humans as point targets based on the fact that radars generally do not measure appearance information, it should be noted that we expect this assumption will need to be relaxed at some point in the future. The reason is that the micro-Doppler signatures from individual humans are distinct as well and thus have biometric potential [10, 62, 184]. This work is still in its infancy but would obviously be a boon to data association if such information were included in the measurements. It remains to be seen whether it can be made to be reliable under realistic conditions.

We could only find a few relevant examples in the literature of tracking algorithms for outdoor surveillance of human targets using radar. Most of the work based on this combination of target and sensor was instead focused on detection algorithms and near-range indoor applications. Chang et al. [29] used an ultra-wideband (UWB) monostatic impulse radar to detect and track an unknown number of human targets. They employed a cluster-based variant of the multiple hypothesis tracker (MHT) to perform the data association, where the clusters were used to model the multipath returns that are common with type of radar. This extended the earlier work of these same authors, plus two additional collaborators, that assumed a fixed number of targets [28]. Here, the multipath returns were modeled as a point process and an expectation-maximization (EM) algorithm was used to derive a single measurement vector that could be incorporated into a Kalman filter update. Blackman et al. [12] considered the problem of tracking human targets in a potentially dense clutter background given feature-rich measurements. The features, based on micro-Doppler characteristics, were used to identify measurements originating from a human targets. It was assumed that in addition to false alarm measurements, measurements originating from animals or vehicles might also be present. The data association was performed using the MHT, which is the most common approach for radar ground tracking algorithms of all kinds because of its ability to handle clutter and missed detections.

Although designed with vehicles and state estimation in mind, there are radar ground tracking algorithms that overlap to some extent with aspects of our proposed

approach.

In chapter 4, we take up the question of how to model the dynamics of a highly maneuverable human target. Since vehicles present a similar challenge, it is instructive to see what has been applied to this problem. The overwhelming consensus is to use multiple model estimators with at least two models (e.g. [176, 103, 102, 137, 11]). The first is a low-noise constant velocity model to describe the motion of the target when it is not maneuvering and the second is a high-noise model, of the same form, intended to accommodate maneuvers. The form of these models is equivalent to the Wiener process speed model that we will consider in section 4.3.2. Most algorithms also include an additional zero velocity model to describe the motion (or lack thereof) of a stopped target. This is motivated in part by the need to address the move-stop-move behavior of evasive targets [102]. Shea et al. [176] also included a model to describe "stationary moving" targets, which are those that are rotating but not translating. Kirubarajan et al. [103] adjusted the noise parameters of the models based on the local terrain. Kastella and Kreucher [97] sought to incorporate terrain information directly into the kinematics of the model. This was accomplished by using an inhomogenous integrated Ornstein-Uhlenbeck (IIOU) process model to describe a position-dependent preferred speed and heading. These preferences could be learned from GPS truth or "hospitality for maneuver" (HM) data which characterizes the ease with which a vehicle can traverse a particular area. They employed a default constant velocity constant heading model when the preferred speed and heading is not available.

We will adopt a multiple model approach based on "stop" and "go" models that is similar to these vehicle tracking algorithms. For stopped targets, we will use the standard zero-velocity model. To model moving targets, we will evaluate the effectiveness of several different types of speed model, including an Ornstein-Uhlenbeck process. Because of our assumption that the targets are sparsely observed, we will consider a different means of accommodating target maneuvers. Instead of including a high-noise variant of the "go" model, we propose to use the current measurement to infer the parameters of the model. This is based on the assumption that the maneuver time is much less than the sample period so changes to the target's trajectory could potentially be quite significant. The usual assumption is that such changes occur over the span of multiple measurements so the uncertainty in each state estimate need not be excessively large. We can justify looking ahead to set the model based on the measurement because

the state estimation is performed in the context of classification rather than prediction.

The movement of vehicles is constrained by roads and terrain so it makes sense to incorporate as much of this information as possible into the motion models. Of these two constraints, the influence of the road network is the most amenable to practical, especially Gaussian, models. Consequently, almost all vehicle tracking algorithms apply road-based constraints to improve their performance. This has been done by adjusting the process noise covariance so that it is aligned with the direction of the road [103], treating the road information as a pseudo-measurement [52], and using a road-based coordinate system [11, 105]. Of course, vehicles do not always confine their movements to roads (or known roads). Thus, practical systems must account for off-road movement as well by including unconstrained motion models in the model set. However, the fact that vehicles may travel both on-road and off-road presents us with a dilemma. If we rely too much on the road information, then we might end up with a biased location estimate or introduce significant errors when the target is not following the road. On the other hand, failing to apply these constraints can also degrade the performance. Finding the right balance can be difficult. Shea et al. [176] tried to reliably determine when a target was moving under the influence of a road by modeling the width of the road as a Gaussian sum of ellipses to allow for easy comparison with the position covariance. Pannetier et al. [137] used a fuzzy detector to model the uncertainty about the influence of the road network. Branching within the road network presents another challenge. Pannetier et al. used a sequential probability ratio test to select the road chosen by the target. Kirubarajan et al. [103] introduced an additional model for each branch into a variable structure multiple model estimator.

While the movement of human targets is less constrained by roads/paths and terrain, it still makes sense to try to incorporate this information into the motion models. Similar to the road constraints used by existing vehicle tracking algorithms, we will consider path-constrained dynamic models in section 4.3.6. Note that the term path is used generically here and may refer to anything from a paved road to a narrow dirt trail. Regarding the question of how much to rely on this information, we will regard the path constraints as soft constraints designed to express a preference for path following while still accommodating off-path motion. The idea is that paths represent favorable terrain and thus targets prefer to move along those routes but are not bound by them. We also want to accommodate behavior such as movement parallel to the path. In the manner

of Shea et al., a radius from the path centerline (not necessarily symmetrical) defines the influence of the path which may be set based on terrain considerations rather than strictly representing the width of the path itself. Branching is handled by introducing an additional model for each branch into a variable structure multiple model estimator as was done by Kirubarajan et al.

Terrain obscuration is a well-known complication of ground target tracking. In regions where the target cannot be seen by the sensor, the probability of detection (P_D) drops to zero. Several algorithms incorporated this information into their models and we will do the same. Kirubarajan et al. [103] recorded the visibility of each road segment. When a segment with no visibility was used by the motion model, the state estimate was made without the measurement update. In later work, Kirubarajan and Bar-Shalom [102] used an ad hoc modification of each likelihood in the multiple model estimator to account for P_D. For the moving target models, the end effect was equivalent to using our augmented emission probability (equation (2.15)) under the assumption the the P_D is not a function of the state. For the stopped target model, they assumed that the P_D was zero because the target was below the minimum detectable velocity of the GMTI. Although we do not specifically address the problem of missed detections of stopped targets, our assumption that the P_D is a function of the target state means that we could implement such behavior if necessary. Kastella and Kreucher [97] considered the problem of tracking targets using nonlinear filtering given amplitude measurements defined over a spatial grid. They compared the performance of measurement models based on the raw amplitudes and those based on applying detection threshold. The first case represents a track-before-detect framework where the measurement model can naturally account for the effects terrain obscuration since it describes the amplitude of the received signal (see [179] for more details). In the second case, they do in fact model a spatially dependent P_D by defining the probability of the detection event but not in a way that generalizes to other measurement models. Our approach of using a measurement space augmented to represent the possibility of no detection (see sections 2.2 and 2.3.1) is intended to provide a means of incorporating a spatially dependent P_D into a standard measurement model.

1.4 Outline of the Dissertation

Chapter 1 Introduces the notion that state estimation and data association are two distinct, but related, objectives of tracking that each require a different type of model to capture the relevant uncertainty. Wide-area surveillance is used as one of the motivating applications for tracking as a data association problem. The technical background and related work are discussed.

Chapter 2 Presents the associative tracking model which represents a direct and general purpose solution to the multiple-scan data association problem. Two different algorithms based on Monte Carlo sampling are proposed for performing inference on the model.

Chapter 3 Takes a detailed look at possible models for the prior distribution of the cluster labels when the number of clusters is unknown. This includes identifying the most promising existing models and proposing several novel extensions.

Chapter 4 Considers the application of the associative tracking model to the wide-area surveillance of human targets using radar. A study is conducted to learn about the motion characteristics of human walkers observed at a distance and on a coarse time-scale. The insights are then used to develop a variety of possible dynamic models.

Chapter 5 Evaluates the dynamic models using simulated measurements generated from real target trajectories. Using the top performing dynamic model to define the likelihood, the sample obtained from the associative tracking model is evaluated for different combinations of cluster label prior and inference algorithm. The evaluations are performed using both simulated measurements and radar data.

Chapter 6 Discusses the results of this work and suggests areas for further research.

Chapter 2

Associative Tracking Model

2.1 Introduction

The purpose of the model presented here is to explicitly address the multiple target tracking problem as fundamentally one of data association, or equivalently, pattern recognition. Such an approach is necessary when the objective is to identify and interpret those patterns in the data that indicate the presence of targets in order to develop an understanding of what is happening in the surveillance region. A substantial body of work already exists in which tracking is treated as fundamentally a state estimation problem. This type of approach is appropriate when the objective is to provide the single best guess as to what is happening, in the form of more precise, or more comprehensive, estimates of each target's state than could be obtained directly from the measurements.

As described in the previous chapter, the two perspectives of tracking are intimately related, yet require different models and inference. Due to this relationship, some existing work is compatible with the pattern recognition view of tracking. The associative tracking model expands on this work to provide a complete solution, including efficient inference, for single-sensor multiple target tracking problems in which there are maneuvering targets, sparse measurements, significant numbers of false alarms, and a spatially varying probability of detection that may be $\ll 1.0$.

The model is designed to exploit the overlap between state estimation and data association by internally employing a state estimation model, but then integrating over all possible states. This enables the use of a large and familiar class of existing target dynamic models.

2.2 Preliminaries

A single sensor is assumed to monitor the region under surveillance. The sensor data from each scan is processed by a detector that outputs a measurement whenever a target is deemed to be present.

Let the vector $\mathbf{y}_i \in \mathbb{R}^m$ be the i-th measurement output by the detector. Each element corresponds to one of the measured values and is denoted by $(\mathbf{y}_i)_j$, where $1 \leq j \leq m$. At a minimum, the measured values are assumed to include the target's position.

The $\mathbf{y}_i$ are points in the the space of "positive" measurements $\mathcal{M} \subseteq \mathbb{R}^m$. These

are the physical measurements generated during each scan of the sensor in response to the occurrence of a detection event (i.e. the received signal exceeds some threshold inside of the detector so that the presence of a target is declared). The qualifier "positive" is used to indicate that such measurements imply the occurrence of a corresponding detection event. This is in contrast to the notion of "negative" measurements, which imply the nonoccurrence of a detection event, or equivalently, the occurrence of a no detection event. The terminology comes from the fact that the information conveyed through knowledge of a no detection event is commonly referred to as "negative information".

Negative measurements are not explicitly reported by the detector, but instead, are inferred from the lack of any physical measurement output by the detector when analyzing the signal collected at a given point in space and time. Since a no detection event is observable, it can be processed just like a physical measurement, given a likelihood model that describes the probability of detection. The probability of detection, P_D, is the probability of the detector declaring the presence of a target when a target is in fact present.

For some applications, P_D can reasonably be assumed to be constant and equal to 1.0. This implies that all targets present within the surveillance region are detected in each scan and no detection events only occur in the absence of a target. Under these circumstances, the negative information is redundant since the detection process is essentially deterministic. Thus, the positive measurements can be used directly (without qualification) and the negative measurements can simply be ignored. Most existing tracking models do just that. However, for the applications motivating this research, P_D is assumed to vary spatially, even dropping to zero in some locations due to a lack of visibility to the sensor. Under these circumstances, negative information is in fact useful and should be incorporated into the tracking model. This can be accomplished by defining an augmented measurement space that encompasses both positive and negative measurements.

Let the augmented measurement space be $\tilde{\mathcal{M}} = \mathcal{M} \cup \emptyset$, where $\mathcal{M}$ is the space of real-valued physical measurements and $\emptyset$ is a discrete element that indicates the absence of a physical measurement. The same formulation was used by Särkkä et al. in the context of multiple sensors [169]. Thus, the measurements $\tilde{\mathbf{y}}_i \in \tilde{\mathcal{M}}$ are actually mixed random variables in which the continuous part represents the outcomes from a positive measurement and the discrete part represents the trivial (empty) outcome from

a negative measurement. The probabilities defined over the continuous part describe our beliefs about the joint occurrence of a detection event and the resulting value of the physical measurement. The probability defined over the discrete part describes our beliefs about no target being detected and hence no physical measurement. When its value is known, we will refer to $\tilde{\mathbf{y}}_i = \emptyset$ as a negative measurement and $\tilde{\mathbf{y}}_i \neq \emptyset$ as a positive measurement. Note that any measurement output by the detector is necessarily a positive measurement since a detection event is required to trigger the output. Again, negative measurements must be inferred from our knowledge of other events. For convenience, we will use $\mathbf{y}_i$ to refer to both a physical measurement and the continuous part of any corresponding $\tilde{\mathbf{y}}_i$.

Let the set of all measurements under consideration be $\tilde{Y} = \{\tilde{\mathbf{y}}_i : 1 \leq i \leq nm\}$, where nm is the number of measurements output by the detector throughout a time window of duration $T \in \mathbb{R}_{>0}$ seconds. These measurements are positive by definition and together they constitute the observed data. The indices are assumed to be defined such that the measurements are partially ordered with respect to scan. In other words, for $i < j$, the measurement $\tilde{\mathbf{y}}_j$ was generated during the same sensor scan as $\tilde{\mathbf{y}}_i$ or during a subsequent scan.

The subset of $\tilde{Y}$ containing the measurements generated during scan i is denoted by $\tilde{Y}_{s_i} = \left\{\tilde{\mathbf{y}}_{s_i(j)} : 1 \leq j \leq nm_{\mathbf{s}_i}\right\}$, where $nm_{\mathbf{s}_i}$ is the number of measurements in the scan. The subscript $s_i(j)$ should be understood as mapping the j-th measurement in the scan i to its natural index in $[1..nm]$. With respect to j, the ordering of measurements within a scan shall be considered arbitrary to avoid introducing any dependencies on the way the sensor conducts a scan or the order in which the detector processes the data. These subsets form a partition of the measurements such that $\tilde{Y} = \bigcup_{i=1}^{ns} \tilde{Y}_{s_i}$, where ns is the number of scans and $\tilde{Y}_{s_a} \cap \tilde{Y}_{s_b} = \emptyset$ for $a \neq b$. Note that some $\tilde{Y}_{s_i}$ may be empty since detections will not necessarily occur in all scans.

Associated with each measurement, $\tilde{\mathbf{y}}_i$, is a scalar indicator variable, $z_i \in \mathbb{Z}_{\geq 0}$, that identifies the origin of the measurement. A value of zero indicates that the measurement is believed to have originated from a generic false alarm source. Values greater than zero indicate that the measurement is believed to have originated from a target. All measurements whose respective indicator variables share a common value are deemed to have originated from the same source. Without loss of generality, the target indicator values are assumed to be assigned consecutively so that given nk targets,

$z_i \in [0..nk]$. Let the set of all indicator variables be $Z = \{z_i : 1 \leq i \leq nm\}$.

Note that the indicator variables are defined such that each target measurement is assumed to have a unique origin. Consequently, so-called merged measurements, which occur when the sensor resolution is insufficient to discriminate multiple closely separated targets, cannot be modeled. This is a simplifying assumption that will not always be true in practice; especially for radar sensors monitoring human targets at long range. However, the best way to relax the assumption to support merged measurements remains an open question. We leave the matter to future work because it would entail the research and development of a different class of prior distributions over Z than those that we set out to study here.

The indicator values can be interpreted as cluster labels, where each cluster contains the measurements from a different underlying source. Thus, given Z, the subsets of $\tilde{Y}$ corresponding to these clusters can be defined. Each such subset is denoted by $\tilde{Y}_{\tau_k} = \left\{\tilde{\mathbf{y}}_{\tau_k(j)} : 1 \leq j \leq nm_{\tau_k}\right\}$, where nm_{τ_k} is the number of measurements of target k, when $k \in [1..nk]$, or the number of false alarm measurements, when $k = 0$. The subscript $\tau_k(j)$ should be understood as mapping the j-th measurement associated with the indicator value k to its natural index in $[1..nm]$. With respect to j, the measurements are time-ordered. These subsets form a partition of the measurements such that $\tilde{Y} = \bigcup_{k=0}^{nk} \tilde{Y}_{\tau_k}$ and $\tilde{Y}_{\tau_a} \cap \tilde{Y}_{\tau_b} = \emptyset$ for $a \neq b$. The subset of false alarm measurements, $\tilde{Y}_{\tau_0}$, may be empty. The subsets of target measurements, however, cannot be empty by definition; they represent the tracks.

Since we know the scan in which each measurement was generated, the associations between targets and measurements conveyed by Z also tells us the scans in which each target was detected. This is important because it implies that when conditioned on Z, we can infer the scans in which the target was not detected, and hence infer negative measurements. A target is assumed to persist in the surveillance region from the time of the first scan in which it was detected to the time of the last scan in which it was detected. Between these two scans, any missed detections imply the presence of a negative measurement in their respective scans. Let ns_{τ_k} be the number of scans in the lifetime of target k.

It is assumed that in each scan, any given target is detected at most one time. One implication of this assumption is that so-called split measurements, which occur when a single target triggers multiple detections during the same scan, cannot be

modeled. This presents a problem for certain sensor modalities, but is not expected to be a serious limitation in practice. Another implication is that of a scan constraint on the cluster labels. If a target can be detected at most one time during each scan, then $\tilde{Y}_{\tau_k}$ may not contain more than one measurement from each scan, $k \in [1..nk]$.

Implicitly associated with each measurement, $\tilde{\mathbf{y}}_i$, is a non-random time variable, $t_i \in \mathbb{R}_{\geq 0}$, that records when the measurement was made. For negative measurements, the time is taken to be the time of the last positive measurement plus the appropriate multiple of the scan period, $T_s \in \mathbb{R}_{>0}$. The scan period is defined as the amount of time between consecutive scans of any given location.

Also associated with each measurement, $\tilde{\mathbf{y}}_i$, is a binary indicator variable, $z0_i \in \{0, 1\}$, that specifies whether or not the measurement is a false alarm. This, of course, is redundant as $z0_i$ is equivalent to $\mathbb{1}(z_i = 0)$. However, the dedicated false alarm indicator variables are needed in order to define the false alarm model. The tracking model itself will be defined such that the joint probability of $z0_i$ and z_i is zero whenever their values conflict (i.e. $z_i > 0$ when $z0_i = 1$, or $z_i = 0$ when $z0_i = 0$). Let the set of all false alarm indicator variables be $Z0 = \{z0_i : 1 \leq i \leq nm\}$.

The subset of $Z0$ containing the false alarm indicator variables associated with the measurements in scan i is denoted by $Z0_{s_i} = \left\{z0_{s_i(j)} : 1 \leq j \leq nm_{s_i}\right\}$. These subsets form a partition of the false alarm indicators such that $Z0 = \bigcup_{i=1}^{ns} Z0_{s_i}$ and $Z0_{s_a} \cap Z0_{s_b} = \emptyset$ for $a \neq b$. Note that some, or all, of the $Z0_{s_i}$ may be empty. The total number of false alarm measurements is given by $nfa = \sum_{i=1}^{nm} z0_i$. The number of false alarm measurements in scan i is given by $nfa_{s_i} = \sum_{j=1}^{nm_{s_i}} z0_{s_i(j)}$.

Collectively, the indicator variables in Z and $Z0$ represent an interpretation of the observed data. Equivalently, they can be viewed as representing a solution to the data association problem. They encode all of the information relevant to our understanding of what is happening within the surveillance region. Therefore, in order to achieve the ultimate objective, the associative tracking model must describe the probability of such an interpretation, given the observed data: $p(Z, Z0 \mid \tilde{Y})$.

By themselves, Z and $Z0$ provide just a basic summary of what is happening in the surveillance region, including the number of targets and when they were detected. However, given a particular instance of these variables, the respective one-to-one relationships between the measurements and the two sets of indicators, Z and $Z0$, mean that a detailed account of the situation, including where and how the targets are mov-

ing, can be readily derived. When the inherent measurement precision is insufficient to be used directly to describe the targets' behavior, or additional unmeasured attributes are required, the mapping of measurements to targets conveyed by Z and $Z0$ enables us to set up a series of well-posed state estimation problems from which the desired information can be obtained.

That last point is important because it highlights the key distinction between associative tracking and conventional tracking; it also illustrates the relationship between the two approaches. Associative tracking models the space of interpretations of the data. This enables us to infer the most likely interpretations and evaluate different interpretations in order to visualize them or their quantify their uncertainty. Any interpretation of interest can then be converted to a set of state estimates, when necessary, to refine our understanding of the situation.

Conventional tracking, on the other hand, models the state space of each individual target. Of course, this requires an interpretation of the observed data because the number of targets and their lifetimes must be known in order to assign a state model to each target. In practice, the required interpretation is built up sequentially in conjunction with the inference of the most likely target states at each time step. The appearance and disappearance of targets is determined using likelihood ratio tests. Such external track management is needed to maintain the set of active target state models. Uncertainty in the relationships between measurements and targets is dealt with either by estimating the most likely associations or by approximately integrating over all possible associations, both conditioned on the past and present states of the active targets. Thus, the output reflects just a single interpretation of the observed data. Such an approach makes sense when we can safely assume that the distribution of interpretations has a single dominant mode. Unfortunately, it is difficult to know when that assumption is warranted and the set of state estimates produced by the tracker sheds no light whether that is in fact the case.

2.3 Model Definition

Associative tracking is based on the following hierarchical model:

$$p\Big(\lambda \mid a_\lambda,\, b_\lambda\Big) \sim \Gamma\Big(a_\lambda,\, b_\lambda\Big) \tag{2.1}$$

$$p\Big(\mathit{Z0} \mid \lambda\Big) \sim f_{\mathit{Z0}}\Big(\mathit{Z0};\, \lambda\Big) \tag{2.2}$$

$$p\Big(Z \mid \mathit{Z0},\, \alpha\Big) \sim f_Z\Big(Z;\, \mathit{Z0},\, \alpha\Big) \tag{2.3}$$

$$p\Big(\alpha \mid \mathit{Z0}\Big) \sim \Gamma\Big(a_\alpha(\mathit{Z0})\,,\, b_\alpha(\mathit{Z0})\Big) \tag{2.4}$$

$$p\Big(\tilde{Y} \mid Z,\, \mathit{Z0}\Big) \sim f_{\tilde{Y}}\Big(\tilde{Y};\, Z,\, \mathit{Z0}\Big)\,. \tag{2.5}$$

Besides the indicator variables in Z and $\mathit{Z0}$ that represent the requisite interpretation of the observed data, the model contains two additional variables, λ and α, that will also be included in the inference process.

The first variable, λ, is a hyperparameter of the prior distribution over false alarm indicators, $p(\mathit{Z0} \mid \lambda)$. It is the false alarm rate in units of false alarms per scan. The exact false alarm rate is assumed to be unknown so λ is modeled as a random variable distributed according to the hyperprior $p(\lambda \mid a_\lambda,\, b_\lambda)$, with known parameters a_λ and b_λ.

The second variable, α, is a hyperparameter of the prior distribution over cluster labels, $p(Z \mid \mathit{Z0},\, \alpha)$. For a certain class of priors, which we will examine in detail in the next chapter, α is a concentration parameter which controls the formation of new clusters. It too is assumed to be unknown and modeled as a random variable. The concentration parameter is distributed according to the hyperprior $p(\alpha \mid \mathit{Z0})$, with parameters that are a function of the random false alarm indicator variables. For cluster label priors without any hyperparameters or with known hyperparameters, (2.4) drops out of the model and any known hyperparameters take the place of α in the conditioning in (2.3).

This is a generative model as it describes the joint distribution of the observed data and the unknown variables. Therefore, Baye's rule is used to calculate the desired conditional:

$$p\Big(Z,\, \mathit{Z0},\, \lambda,\, \alpha \mid \tilde{Y},\, a_\lambda,\, b_\lambda\Big) \propto$$
$$p\Big(\tilde{Y} \mid Z,\, \mathit{Z0}\Big)\, p\Big(Z \mid \mathit{Z0},\, \alpha\Big)\, p\Big(\alpha \mid \mathit{Z0}\Big)\, p\Big(\mathit{Z0} \mid \lambda\Big)\, p\Big(\lambda \mid a_\lambda,\, b_\lambda\Big)\,, \tag{2.6}$$

which is the joint posterior distribution—up to the normalizing constant—of the cluster labels, false alarm indicators, false alarm rate, and concentration parameter (when applicable), given the observed data and known parameters. The likelihood and each component of the prior will be described in detail in the following subsections.

Ideally, we would like a complete characterization of the joint posterior distribution. A point estimate, even if it can be assumed to correspond to the maximum a posteriori (MAP) probability, conveys no information about its uniqueness or whether it is representative of the majority of the distribution. This is insufficient given the objective of understanding what is happening in the surveillance region. In this context, knowledge about alternative interpretations of the data can be critical; especially those that are nearly as likely as the mode.

Relying on a point estimate would, in fact, undermine much of the original motivation for associative tracking. However, even given circumstances under which a point estimate would be sufficient, there might be some utility in using associative tracking so that the confidence of the estimate can be quantified.

With that said, what exactly is meant by "a complete characterization"? That will depend to a large extent on the specific application. The dimensionality of the joint posterior grows with the number of measurements so, in general, it can be regarded as high-dimensional. This means that there is no straightforward way of visualizing the distribution nor are there any summary statistics that will make sense in all cases. The requirements of the particular application will dictate what information about the distribution is most important and what form it must take to be amenable to processing or display.

To facilitate such application-specific inference, we need a sample of points drawn from the joint posterior. Such a sample can then be used to compute Monte Carlo estimates of arbitrary functions, with respect to the distribution. In other words, for some function of the unknown variables, $g(Z,\, Z\!0,\, \lambda,\, \alpha)$, we can compute summary statistics of the form:

$$\mathbb{E}\left[g\Big(Z,\, Z\!0,\, \lambda,\, \alpha\Big)\right] = \int g\Big(Z,\, Z\!0,\, \lambda,\, \alpha\Big)\, p\Big(Z,\, Z\!0,\, \lambda,\, \alpha \mid \tilde{Y},\, a_\lambda,\, b_\lambda\Big)\, dZ\, dZ\!0\, d\lambda\, d\alpha.$$

Since the joint posterior is not of a form which allows for direct sampling, Monte Carlo methods can be used to generate a sample from the distribution. This requires that $p(Z,\, Z\!0,\, \lambda,\, \alpha \mid \tilde{Y},\, a_\lambda,\, b_\lambda)$ be evaluated only up to the normalizing constant so we can

work directly with (2.6), which is a significant computational savings. The high dimensional parameter space renders exact computation of the normalizing constant intractable and reliable approximations difficult. Although MCMC spares us this expense, one consequence of the dimensionality that we cannot avoid is that it makes it significantly more challenging to construct a sampler that explores the space efficiently. In the next section, we will present a sampling algorithm designed to meet this challenge.

2.3.1 Likelihood

The likelihood is given by equation (2.5) in the hierarchical model. It is the joint distribution of the observed, hence positive, measurements given the interpretation of the data represented by the indicator variables. The probability density function is denoted by $f_{\tilde{Y}}(\tilde{Y};\, Z,\, \mathcal{Z}0)$ to indicate that it does not match any well-known distribution.

To define the density of the likelihood, we begin with the assumption that the measurements originating from each source (i.e. in each cluster) are independent. Thus, the joint distribution of the measurements, given Z and $\mathcal{Z}0$, can be factored as:

$$p\Big(\tilde{Y} \mid Z,\, \mathcal{Z}0\Big) = \prod_{k=0}^{nk} p\Big(\tilde{Y}_{\tau_k} \mid Z,\, \mathcal{Z}0\Big)\,, \tag{2.7}$$

which is the product of the false alarm measurement likelihood and the nk target measurement likelihoods.

This independence assumption represents a tradeoff. It makes the model is far more tractable, but unable to describe interactions between targets. Some models capable of dealing with target interactions have been developed (e.g. [100]), however, it is not clear how best to apply them to associative tracking. The typical targets of interest in associative tracking applications are goal-seeking and intent-driven. As such, in some contexts we might expect to see evasive interactions and in others, cooperative interactions for the purpose of achieving some objective. It is left to future work to develop useful models of this kind of complex high-level behavior.

The false alarm measurement likelihood corresponds to the product term $k = 0$ in (2.7). This is the joint distribution of the false alarm measurements, given Z and $\mathcal{Z}0$. False alarms are assumed to occur independently, so the joint distribution can be factored as:

$$p\Big(\tilde{Y}_{\tau_0} \mid Z,\, \mathcal{Z}0\Big) = \prod_{j=1}^{nm_{\tau_0}} p\Big(\tilde{\mathbf{y}}_{\tau_0(j)} \mid Z,\, \mathcal{Z}0\Big)\,,$$

where each of these product terms describes the probability of an individual false alarm measurement.

Typically, $p(\tilde{\mathbf{y}}_{\tau_0(j)} \mid Z, \mathcal{Z}0)$ is defined as a uniform distribution over the measured values. For position measurements, the support of distribution is set to match the sensor coverage. For other other measured values, such as Doppler velocity, the support of the distribution is set to match the range of feasible values. Note that false alarm measurements are always positive measurements so $p(\tilde{\mathbf{y}}_{\tau_0(j)} = \emptyset \mid Z, \mathcal{Z}0) = 0$, for $1 \leq j \leq nm_{\tau_0}$.

When using a radar with a fixed threshold detector, false alarms will only be spatially uniform when detecting targets against a background of spatially uncorrelated clutter. Most detectors, however, use an adaptive algorithm to adjust the detection threshold so that it maintains a constant false alarm rate (CFAR). An ideal CFAR detector will have a spatially uniform probability of false alarm and hence a spatially uniform distribution of false alarms. This justifies starting with a uniform distribution over the position measurements in most cases. In practice, the clutter power will not always be perfectly modeled and some parts of the environment will trigger more false detections than others. This situation can be identified experimentally and an appropriate false alarm measurement distribution learned from the data. Fortunately, there are no restrictions on the distributional form of $p(\tilde{\mathbf{y}}_{\tau_0(j)} \mid Z, \mathcal{Z}0)$; it need only be evaluated.

The target measurement likelihoods correspond to the product terms $1 \leq k \leq nk$ in (2.7). Each term is the joint distribution of the measurements of target k, given Z and $\mathcal{Z}0$. To define that distribution, we begin by expanding the conditioning to include any negative measurements that are implied by the knowledge of Z. As discussed in the preliminaries, the life of a target is assumed to span the first scan in which it was detected to last scan in which it was detected. Any missed detections of that target in the intervening scans constitute negative measurements. Let the set of negative measurements associated with target k be $\tilde{Y}'_{\tau_k}$, which may be empty in the event that there are no missed detections of the target. Thus, the target measurement likelihood

can be equivalently written as:

$$
\begin{aligned}
p\Big(\tilde{Y}_{\tau_k} \mid Z,\, Z0\Big) &= p\Big(\tilde{Y}_{\tau_k} \mid Z,\, Z0,\, \tilde{Y}'_{\tau_k}\Big) \\
&= \frac{p\Big(\tilde{Y}_{\tau_k},\, \tilde{Y}'_{\tau_k} \mid Z,\, Z0\Big)}{\int p\Big(\tilde{Y}_{\tau_k},\, \tilde{Y}'_{\tau_k} \mid Z,\, Z0\Big)\, d\tilde{Y}_{\tau_k}} \\
&= \frac{p\Big(\tilde{Y}_{\tau_k},\, \tilde{Y}'_{\tau_k} \mid Z,\, Z0\Big)}{p\Big(\tilde{Y}'_{\tau_k} \mid Z,\, Z0\Big)},
\end{aligned} \tag{2.8}
$$

where the last two lines are the result of applying Baye's rule to rewrite the conditional distribution of the positive measurements as the joint distribution of the positive and negative measurements, divided by the marginal distribution of the negative measurements. Since the marginal distribution of the negative measurements is derived directly from the joint distribution of the positive and negative measurements, let us proceed by defining the joint distribution.

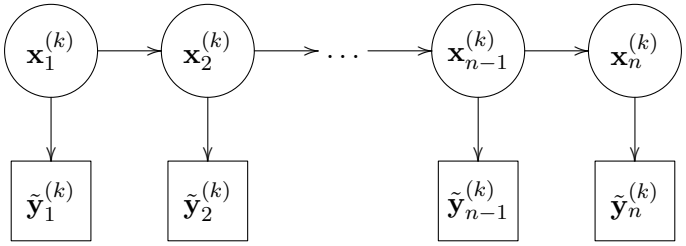
For a given target, there are statistical dependencies among the measurements because the target is undergoing coherent motion. To model these dependencies, we introduce a set of latent variables that represent the target's state. Let $X^{(k)} = \left\{\mathbf{x}_j^{(k)} : 1 \leq j \leq ns_{\tau_k}\right\}$ be the set of state variables corresponding to target k, where $\mathbf{x}_j^{(k)} \in \mathbb{R}^n$ represents the state of target in the j-th scan of its lifetime. For convenience, the state vectors are indexed by scan using a relative scan index for which 1 refers to the first scan in which the target was detected and ns_{τ_k} refers to the last scan in which it was detected. From each of these scans, comes one of the measurements of the target. To make the correspondence between scans, state variables, and measurements explicit, let the permutation of $\left\{\tilde{Y}_{\tau_k},\, \tilde{Y}'_{\tau_k}\right\}$ so that its elements are in scan order be denoted by $\tilde{Y}^{(k)} = \left\{\tilde{\mathbf{y}}_j^{(k)} : 1 \leq j \leq ns_{\tau_k}\right\}$. Thus, $\tilde{Y}^{(k)}$ is the set all measurements, both positive and negative, of target k and its elements correspond one-to-one to the state variables in $X^{(k)}$.

The state of the target is assumed to evolve over time as a Markov process. Note that in our notation the relative scan index is being used as a proxy for time, which is known for each measurement. The Markovian assumption implies that $p(\mathbf{x}_j^{(k)} \mid \mathbf{x}_{j-1}^{(k)}, \mathbf{x}_{j-2}^{(k)}, \ldots) = p(\mathbf{x}_j^{(k)} \mid \mathbf{x}_{j-1}^{(k)})$. At each scan, the latent target state is assumed to emit (i.e. probabilistically generate) the observed measurement. Together, these assumptions define a generative model in which the joint distribution of target

states and measurements has the form:

$$p\Big(\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_n^{(k)}, \tilde{\mathbf{y}}_1^{(k)}, \ldots, \tilde{\mathbf{y}}_n^{(k)}\Big) = p\Big(\tilde{\mathbf{y}}_1^{(k)}, \ldots, \tilde{\mathbf{y}}_n^{(k)} \mid \mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_n^{(k)}\Big)\, p\Big(\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_n^{(k)}\Big)$$
$$= p\Big(\tilde{\mathbf{y}}_1^{(k)} \mid \mathbf{x}_1^{(k)}\Big)\, p\Big(\mathbf{x}_1^{(k)}\Big) \prod_{j=2}^{n} p\Big(\tilde{\mathbf{y}}_j^{(k)} \mid \mathbf{x}_j^{(k)}\Big)\, p\Big(\mathbf{x}_j^{(k)} \mid \mathbf{x}_{j-1}^{(k)}\Big), \tag{2.9}$$

where $n = ns_{\tau_k}$ and the conditioning on Z and $Z0$ is left unstated to uncluttered the notation. This factorization is equivalent to the following graphical model:



where the circles represent the latent state variables and the squares represent the observed measurement variables.

Using this model, we can obtain the joint distribution of the positive and negative measurements of target k, given Z and $Z0$, by integrating out the latent states. Therefore, we have:

$$p\Big(\tilde{Y}_{\tau_k},\ \tilde{Y}'_{\tau_k} \mid Z,\ Z0\Big) = \int p\Big(\tilde{Y}^{(k)},\ X^{(k)} \mid Z,\ Z0\Big)\, dX^{(k)}$$
$$= \int \cdots \int p\Big(\mathbf{x}_1^{(k)}\Big)\, p\Big(\tilde{\mathbf{y}}_1^{(k)} \mid \mathbf{x}_1^{(k)}\Big)\, p\Big(\mathbf{x}_2^{(k)} \mid \mathbf{x}_1^{(k)}\Big)\, p\Big(\tilde{\mathbf{y}}_2^{(k)} \mid \mathbf{x}_2^{(k)}\Big) \cdots$$
$$p\Big(\mathbf{x}_n^{(k)} \mid \mathbf{x}_{n-1}^{(k)}\Big)\, p\Big(\tilde{\mathbf{y}}_n^{(k)} \mid \mathbf{x}_{n-1}^{(k)}\Big)\, d\mathbf{x}_1^{(k)} \cdots d\mathbf{x}_n^{(k)} \tag{2.10}$$

Whether or not this multiple integral can be solved analytically depends on how we define $p(\mathbf{x}_j^{(k)} \mid \mathbf{x}_{j-1}^{(k)})$ and $p(\tilde{\mathbf{y}}_j^{(k)} \mid \mathbf{x}_j^{(k)})$, the state transition distribution and emission distribution, respectively. When each of these distributions is defined as a Gaussian whose variables are linearly related, or proportional to such a distribution, the integral can be solved exactly in linear time. More generally, a linear-time solution exists when the distributions are part of the exponential family [126]. For arbitrary distributions, sequential Monte Carlo methods (i.e. particle filters) can be used to approximate (2.10) but are much more computationally expensive. The computational demands for this calculation are a major concern because it will need to be performed, for each target, repeatedly during the inference process. For this reason, we will define the state transition and emission distributions using linear-Gaussian models. In the next subsection,

we will discuss how the unscented transform (UT) can be used to efficiently generate robust Gaussian approximations to non-linear and non-Gaussian models.

The evolution of the target's state can thus be modeled as a discrete-time linear system driven by Gaussian noise, which is represented by the following first-order difference equation:

$$\mathbf{x}_j^{(k)} \quad = \quad F_j \mathbf{x}_{j-1}^{(k)} + u_j + w_j, \tag{2.11}$$

where $F_j \in \mathbb{R}^{n \times n}$ is the system matrix, $u_j \in \mathbb{R}^n$ is a known control input, and $w_j \in \mathbb{R}^n$ is the Gaussian process noise. The process noise is assumed to be independent of the state and zero mean with covariance Q_j. The subscripts on system matrix, control input, and noise covariance indicate that the system may be non-stationary. This model can be recognized as the dynamical state model used by the Kalman filter [6, 95]. The equivalent state transition distribution is:

$$p\Big(\mathbf{x}_j^{(k)} \mid \mathbf{x}_{j-1}^{(k)}\Big) \sim N\Big(F_j \mathbf{x}_{j-1}^{(k)} + u_j,\, Q_j\Big)\,. \tag{2.12}$$

The time series of physical measurements generated by the states can similarly be modeled as a discrete-time linear system driven by Gaussian noise, which is represented by the following first-order difference equation:

$$\mathbf{y}_j^{(k)} = H_j \mathbf{x}_j^{(k)} + v_j, \tag{2.13}$$

where $\mathbf{y}_j^{(k)} \in \mathbb{R}^m$ is the physical measurement of the target generated during the j-th scan, $H_j \in \mathbb{R}^{m \times m}$ is the measurement matrix, and $v_j \in \mathbb{R}^m$ is the Gaussian measurement noise. The measurement noise is assumed to be independent of the state and zero mean with covariance R_j. The subscripts on measurement matrix and noise covariance indicate that this system too may be non-stationary. This model may also be familiar as it corresponds to the measurement model used by the Kalman filter. Here, however, it is just one piece of the measurement model since (2.13) only describes how the physical measurements are generated from the target's state. Recall that in order to make use of the negative information conveyed by any missed detections, we are using augmented measurements defined on a sample space that covers both the physical measurements resulting from a detection and the failure to detect the target. The measurement model for the augmented measurements must therefore describe both the detection process and the generation of the physical measurements.

Let the emission distribution for the physical measurements equivalent to (2.13) be denoted by:

$$p^{+}\left(\mathbf{y}_j^{(k)} \mid \mathbf{x}_j^{(k)}\right) \sim N\left(H_j \mathbf{x}_j^{(k)},\, R_j\right). \tag{2.14}$$

Since the continuous part of $\tilde{\mathbf{y}}_j^{(k)}$ represents the joint occurrence of a detection event and the resulting value of the physical measurement, and the discrete part represents the occurrence of a no detection event, the emission distribution for the augmented measurements can be defined as:

$$p\left(\tilde{\mathbf{y}}_j^{(k)} \mid \mathbf{x}_j^{(k)}\right) = \begin{cases} p^{+}\left(\mathbf{y}_j^{(k)} \mid \mathbf{x}_j^{(k)}\right) P_D\left(\mathbf{x}_j^{(k)}\right) & \tilde{\mathbf{y}}_j^{(k)} \neq \emptyset \\ 1 - P_D\left(\mathbf{x}_j^{(k)}\right) & \tilde{\mathbf{y}}_j^{(k)} = \emptyset, \end{cases} \tag{2.15}$$

where P_D is the probability of detecting the target, which is a function of the target's state. Specifically, and without loss of generality, we assume that P_D is a function of the target's position and independent of any other measured values. Note that $P_D(\mathbf{x}_j^{(k)})$ is not a probability distribution, but rather a function whose image is the space of probabilities. In the continuous part of $p(\tilde{\mathbf{y}}_j^{(k)} \mid \mathbf{x}_j^{(k)})$, it is used to scale the probability distribution $p^{+}(\cdot)$ to reflect the conjunction of the detection and physical measurement probabilities. In the discrete part, it is used to compute the remaining probability mass necessary so that the probability of the entire sample space is 1.

When conditioned on the state, the continuous part of (2.15) is proportional to a Gaussian distribution regardless of how $P_D(\mathbf{x}_j^{(k)})$ is defined. However, when the state is unknown and represented as a Gaussian random variable (with unspecified conditioning), the joint distribution of the state and the continuous part of $\tilde{\mathbf{y}}_j^{(k)}$:

$$p\left(\tilde{\mathbf{y}}_j^{(k)},\, \mathbf{x}_j^{(k)} \mid \cdot\right) = p\left(\tilde{\mathbf{y}}_j^{(k)} \mid \mathbf{x}_j^{(k)}\right) p\left(\mathbf{x}_j^{(k)} \mid \cdot\right) \quad \text{for } \tilde{\mathbf{y}}_j^{(k)} \neq \emptyset, \tag{2.16}$$

is not guaranteed to be proportional to a joint Gaussian distribution given an arbitrary P_D model. The form of this distribution is of concern because it impacts whether the integrals in (2.10) have an analytical solution. If it is proportional to a joint Gaussian, then the multiple integral can be solved exactly in linear-time as desired. This can be accomplished either by restricting the form of $P_D(\mathbf{x}_j^{(k)})$ or approximating the true joint distribution using a Gaussian function (i.e. a scaled Gaussian distribution).

One possible way to express the relationship between the probability of detecting a target and its position such that (2.16) has the required form is to define

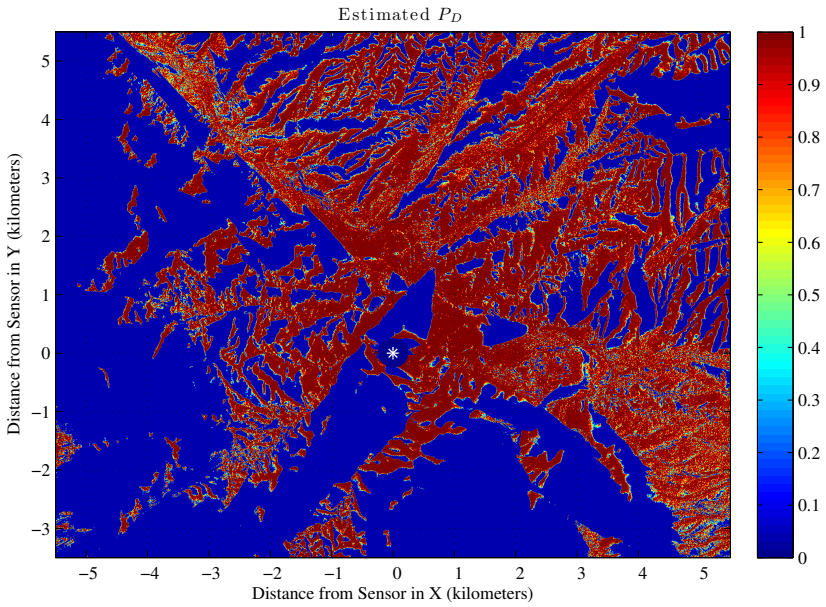


Figure 2.1: Example of a radar P_D model.

$P_D(\mathbf{x}_j^{(k)})$ using a Gaussian function of $\mathbf{x}_j^{(k)}$. This works because the product of Gaussian functions remains a Gaussian function. Unfortunately, the symmetry and form of a Gaussian function is not appropriate for modeling the kind of spatially varying probability of detection that we expect to encounter. For example, figure 2.1 shows a P_D model for a tower mounted surveillance radar in a rugged desert environment. The regions with near-zero probability of detection (dark blue in the figure) correspond to areas with no line-of-sight visibility to the radar. With a nod toward Cromwell's rule, the only zero probability region is a small exclusion zone immediately around the radar where, in fact, no data is recorded; the probabilities in the obscured regions are clamped to a small positive probability to allow for the possibility that the P_D is wrong. The probability of detection in the visible regions was estimated based on clutter models. This irregular patchy structure is typical for ground-based radar P_D models.

Having ruled out a Gaussian P_D model, one simple alternative remains if we wish for $p^+(\mathbf{y}_j^{(k)} \mid \mathbf{x}_j^{(k)})P_D(\mathbf{x}_j^{(k)})$ to remain proportional to a Gaussian distribution so that (2.16) is proportional to a joint Gaussian: a constant P_D model. Not constant across all targets and all scans, since that would undermine the reason why the detection model was incorporated into the likelihood model in the first place, but with a different

value for each combination. Let such a model be defined as:

$$P_D\left(\mathbf{x}_j^{(k)}\right) = pd_j^{(k)},$$

where $pd_j^{(k)} \in [0,1]$ is the probability of detecting target k in the j-th scan in its lifetime. For each target and scan, a suitable value for $pd_j^{(k)}$ can be determined during the process of integrating out the state variables in a manner analogous to linearizing about the current state. When processing the j-th scan in the lifetime of target k, $pd_j^{(k)}$ is set to the expected probability of detection with respect to the distribution of the target's current state, $p(\mathbf{x}_j^{(k)} \mid \cdot)$. This is the optimal value in the sense that it yields the same cumulative probability over the continuous part of (2.15) as the true non-Gaussian distribution. In other words:

$$\int p\left(\tilde{\mathbf{y}}_j^{(k)}\right) d\tilde{\mathbf{y}}_j^{(k)} = \int\int p\left(\tilde{\mathbf{y}}_j^{(k)}, \mathbf{x}_j^{(k)} \mid \cdot\right) d\mathbf{x}_j^{(k)} d\tilde{\mathbf{y}}_j^{(k)} \quad \text{for } \tilde{\mathbf{y}}_j^{(k)} \neq \emptyset,$$

is the same for both constant the constant P_D model and the true P_D model. The physical measurement values, however, will not be distributed in the same way; we will see an example of this discrepancy shortly. In practice, we can simplify the expected probability of detection calculation, at the expense of accuracy, by working with a smoothed representation of $P_D(\mathbf{x}_j^{(k)})$. When minimizing the computational cost is paramount, the P_D model can be smoothed using a kernel that reflects the expected uncertainty in the target state estimates so that computing the value of $pd_j^{(k)}$ is reduced to a single lookup operation.

For an arbitrary P_D model, we can ensure that (2.16) is proportional to a joint Gaussian distribution by explicitly generating such an approximation using samples drawn from the true joint distribution. In this case, the joint distribution of the target's state and the associated measurement is modeled as:

$$p\left(\tilde{\mathbf{y}}_j^{(k)}, \mathbf{x}_j^{(k)} \mid \cdot\right) = \begin{cases} \hat{p}\left(\mathbf{y}_j^{(k)}, \mathbf{x}_j^{(k)}\right) C_j^{(k)} & \tilde{\mathbf{y}}_j^{(k)} \neq \emptyset \\ 1 - C_j^{(k)} & \tilde{\mathbf{y}}_j^{(k)} = \emptyset \end{cases}, \tag{2.17}$$

where:

$$\hat{p}(\mathbf{y}_j^{(k)}, \mathbf{x}_j^{(k)}) \sim N\left(\hat{\mu}_j^{(k)}, \hat{\Sigma}_j^{(k)}\right)$$

is a multivariate Gaussian distribution that describes the relationship between the state and physical measurement, and $C_j^{(k)} \in [0,1]$ is the constant of proportionality. As was done for the constant P_D model, the parameters are estimated, for each target and

scan in its lifetime, during the process of integrating out the state variables. When processing the j-th scan in the lifetime of target k, we can sample from the joint state and measurement distribution by first drawing a set of samples from the distribution of the target's current state, $p(\mathbf{x}_j^{(k)} \mid \cdot)$. Each state sample, $\mathbf{x}'_i$, is then used to generate a measurement sample, $\tilde{\mathbf{y}}'_i$, by making a draw from (2.15), conditioned on $\mathbf{x}'_i$. With probability $P_D(\mathbf{x}'_i)$, the i-th sample is a positive measurement, such that $\tilde{\mathbf{y}}'_i \neq \emptyset$. The measurement values in this case are drawn from $p^+(\mathbf{y}_j^{(k)} \mid \mathbf{x}_j^{(k)})$, conditioned on $\mathbf{x}'_i$. With probability 1-$P_D(\mathbf{x}'_i)$, the i-th sample is a negative measurement, such that $\tilde{\mathbf{y}}'_i = \emptyset$. The mean and covariance are computed from the positive measurement samples. The maximum likelihood estimates of these parameters correspond to the sample mean and sample covariance, respectively. The constant of proportionality is given by the ratio of the number of positive to negative measurements among the samples. One minus this ratio, yields the probability mass of the discrete part of the joint distribution.

It should be clear that the constant of proportionality is in fact an estimate of the expected probability of detection with respect to the distribution of the target's current state. As the number of samples increases, $C_j^{(k)}$ approaches the optimal $pd_j^{(k)}$ value in the constant P_D model. If we replace the constant of proportionality in this model with $pd_j^{(k)}$ and then derive the implied emission distribution, it can be shown that it has the same form as the constant P_D model. Thus, (2.17) actually represents an alternative constant P_D model. This follows from the fact that the conditional distribution of a joint Gaussian is also a Gaussian. In the resulting emission distribution, $\hat{p}(\mathbf{y}_j^{(k)} \mid \mathbf{x}_j^{(k)})$ describes the probability of the physical measurement values and is analogous to $p^+(\mathbf{y}_j^{(k)} \mid \mathbf{x}_j^{(k)})$, but reflects the influence of the probability of detection.

For an example of how the two approaches compare, see figure 2.2. The upper image shows the positional state of a target, with the 6σ extent of its covariance denoted by the ellipse, overlaid on the P_D model (same one as shown in figure 2.1). The most likely positions of the target all lie in a region of near-zero probability of detection, while the tails of the position distribution include regions where the probability of detection is nearly 1. The lower image shows the continuous part of the true marginal (position) measurement distribution computed by integrating over the state in the true joint distribution. It is clearly non-Gaussian because of the influence of the probability of detection. The solid ellipse denotes the extent of the Gaussian marginal measurement distribution computed using the emission distribution for the physical measurements;

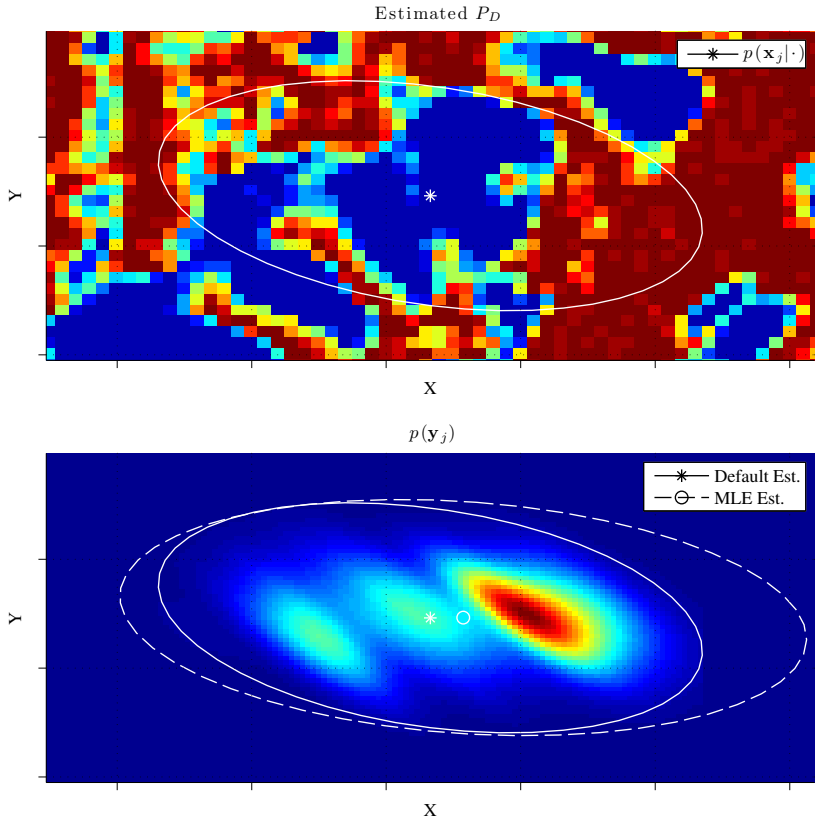


Figure 2.2: Example of a Gaussian approximation to the measurement distribution for an arbitrary P_D model.

without considering the P_D model. The dashed ellipse denotes the extent of the approximated Gaussian marginal measurement distribution computed from samples of the true joint distribution. We can see that the mean has shifted right, toward the largest region with a high probability of detection in the support of $p(\mathbf{x}_j \mid \cdot)$, and the standard deviation in X has increased about 26%. This illustrates that even given a highly non-uniform spatial distribution of P_D, the best Gaussian approximation to the true measurement distribution does not differ significantly from the estimate based on (2.14). It also suggests that it is the scaling factor representing the expected probability of detection that conveys the most important information.

Having defined the state transition distribution and emission distribution, we now return to equation (2.10) and the matter of integrating out the latent states to obtain the joint distribution of the measurements for target k, given *Z* and *Z0*. Let us begin by first considering the objective. After evaluating all the state integrals, we end up with the desired distribution (using the uncluttered notation):

$$p\Big(\tilde{\mathbf{y}}_1^{(k)}, \tilde{\mathbf{y}}_2^{(k)}, \ldots, \tilde{\mathbf{y}}_n^{(k)}\Big),$$

which can be factored using the general product rule for probabilities to get:

$$p\Big(\tilde{\mathbf{y}}_n^{(k)} \mid \tilde{\mathbf{y}}_{n-1}^{(k)}, \ldots, \tilde{\mathbf{y}}_1^{(k)}\Big)\, p\Big(\tilde{\mathbf{y}}_{n-1}^{(k)} \mid \tilde{\mathbf{y}}_{n-2}^{(k)}, \ldots, \tilde{\mathbf{y}}_1^{(k)}\Big) \cdots p\Big(\tilde{\mathbf{y}}_2^{(k)} \mid \tilde{\mathbf{y}}_1^{(k)}\Big)\, p\Big(\tilde{\mathbf{y}}_1^{(k)}\Big).$$

Each of those product terms is a conditional probability that can be obtained by evaluating a single state integral:

$$p\Big(\tilde{\mathbf{y}}_i^{(k)} \mid \tilde{\mathbf{y}}_{i-1}^{(k)}, \ldots, \tilde{\mathbf{y}}_i^{(k)}\Big) = \int p\Big(\tilde{\mathbf{y}}_i^{(k)} \mid \mathbf{x}_i^{(k)}\Big)\, p\Big(\mathbf{x}_i^{(k)} \mid \tilde{\mathbf{y}}_{i-1}^{(k)}, \ldots, \tilde{\mathbf{y}}_1^{(k)}\Big)\, d\mathbf{x}_i^{(k)}, \tag{2.18}$$

for $1 \leq i \leq n$. The first term in the integrand is simply the emission distribution. The second term is the distribution of the target's state in the i-th scan, given all the previous measurements, and can be derived from the output of a Kalman filter.

Given a dynamic model of the form in (2.9), with linear-Gaussian state transition and emission distributions, the Kalman filter provides a recursive procedure for estimating:

$$p\Big(\mathbf{x}_i^{(k)} \mid \tilde{\mathbf{y}}_i^{(k)}, \ldots, \tilde{\mathbf{y}}_1^{(k)}\Big),$$

which is the distribution of the target's state in the scan i, given all measurements up to and including the current. This distribution is Gaussian and thus defined by its mean

and covariance, which are output by the Kalman filter at each time step. Let $\hat{m}_{i|i}$ and $\hat{P}_{i|i}$ be the outputs after processing scans 1 to i so that:

$$p\Big(\mathbf{x}_i^{(k)} \mid \tilde{\mathbf{y}}_i^{(k)}, \ldots, \tilde{\mathbf{y}}_1^{(k)}\Big) \sim N\Big(\hat{m}_{i|i},\, \hat{P}_{i|i}\Big)\,. \tag{2.19}$$

Together, $\hat{m}_{i|i}$ and $\hat{P}_{i|i}$ are referred to as the filtered state estimate.

Conditional on that same information, the target's state in the next scan can be predicted by propagating the filtered state estimate forward in time using the dynamical state model. In terms of the relevant distributions, the prediction is given by:

$$\begin{aligned} p\Big(\mathbf{x}_{i+1}^{(k)} \mid \tilde{\mathbf{y}}_i^{(k)}, \ldots, \tilde{\mathbf{y}}_1^{(k)}\Big) &= \int p\Big(\mathbf{x}_{i+1}^{(k)} \mid \mathbf{x}_i^{(k)}\Big)\, p\Big(\mathbf{x}_i^{(k)} \mid \tilde{\mathbf{y}}_i^{(k)}, \ldots, \tilde{\mathbf{y}}_1^{(k)}\Big)\, d\mathbf{x}_i^{(k)} \\ &= \int p\Big(\mathbf{x}_{i+1}^{(k)} \mid \mathbf{x}_i^{(k)}\Big)\, p\Big(\mathbf{x}_i^{(k)};\, \hat{m}_{i|i},\, \hat{P}_{i|i}\Big)\, d\mathbf{x}_i^{(k)}, \end{aligned} \tag{2.20}$$

which is the state distribution needed to complete the definition of (2.18). This distribution is also Gaussian and given by:

$$p\Big(\mathbf{x}_{i+1}^{(k)} \mid \tilde{\mathbf{y}}_i^{(k)}, \ldots, \tilde{\mathbf{y}}_1^{(k)}\Big) \sim N\Big(\hat{m}_{i+1|i},\, \hat{P}_{i+1|i}\Big)\,.$$

Together, $\hat{m}_{i+1|i}$ and $\hat{P}_{i+1|i}$ are known as the predicted state estimate, which is computed in the course of the Kalman filtering process. From the properties of Gaussian distributions, it can be shown ([6]) that the predicted state estimate for scan $i+1$ is derived from the filtered state estimate in scan i as:

$$\begin{aligned} \hat{m}_{i+1|i} &= F_i \hat{m}_{i|i} + u_i \\ \hat{P}_{i+1|i} &= F_i \hat{P}_{i|i} F_i^T + Q_i. \end{aligned}$$

Once the measurement from scan $i+1$ is received, the Kalman filter is said to update the predicted state estimate by making it conditional on that measurement. The updated prediction is then taken to be the filtered state estimate for scan $i+1$, which is given by:

$$\begin{aligned} p\Big(\mathbf{x}_{i+1}^{(k)} \mid \tilde{\mathbf{y}}_{i+1}^{(k)}, \ldots, \tilde{\mathbf{y}}_1^{(k)}\Big) &= \frac{p\Big(\tilde{\mathbf{y}}_{i+1}^{(k)} \mid \mathbf{x}_{i+1}^{(k)}\Big)\, p\Big(\mathbf{x}_{i+1}^{(k)} \mid \tilde{\mathbf{y}}_i^{(k)}, \ldots, \tilde{\mathbf{y}}_1^{(k)}\Big)}{p\Big(\tilde{\mathbf{y}}_{i+1}^{(k)} \mid \tilde{\mathbf{y}}_i^{(k)}, \ldots, \tilde{\mathbf{y}}_1^{(k)}\Big)} \\ &= \frac{p\Big(\tilde{\mathbf{y}}_{i+1}^{(k)} \mid \mathbf{x}_{i+1}^{(k)}\Big)\, p\Big(\mathbf{x}_{i+1}^{(k)};\, \hat{m}_{i+1|i},\, \hat{P}_{i+1|i}\Big)}{\int p\Big(\tilde{\mathbf{y}}_{i+1}^{(k)} \mid \mathbf{x}_{i+1}^{(k)}\Big)\, p\Big(\mathbf{x}_{i+1}^{(k)};\, \hat{m}_{i+1|i},\, \hat{P}_{i+1|i}\Big)\, d\mathbf{x}_{i+1}^{(k)}}. \end{aligned} \tag{2.21}$$

There are two things to note about these equations. First, they depend on the emission distribution so how we solve them to get $\hat{m}_{i+1|i+1}$ and $\hat{P}_{i+1|i+1}$ will depend on how that

is defined. The standard Kalman filter solution will not apply in all cases. Second, the denominator can be recognized as equal to (2.18), which is precisely what is needed to express the joint measurement distribution.

For the default definition of the emission distribution (equation (2.15)) with the constant P_D model, the standard Kalman filter solution does in fact apply. This is because $p^+(\cdot)$ is equivalent to the emission distribution used by the standard Kalman filter and the terms in $p(\tilde{\mathbf{y}}_{i+1}^{(k)} \mid \mathbf{x}_{i+1}^{(k)})$ related to the probability of detection cancel out between the numerator and denominator. Thus, it can be shown that the updated estimate is derived from the predicted state estimate as:

$$\begin{aligned}\hat{m}_{i+1|i+1} &= \hat{m}_{i+1|i} + K_{i+1}\nu_{i+1} \\ \hat{P}_{i+1|i+1} &= (I - K_{i+1}H_{i+1})\,\hat{P}_{i+1|i},\end{aligned}$$

where:

$$\begin{aligned}\nu_{i+1} &= \mathbf{y}_{i+1}^{(k)} - H_{i+1}\hat{m}_{i+1|i} \\ S_{i+1} &= H_{i+1}\hat{P}_{i+1|i}H_{i+1}^T + R_{i+1} \\ K_{i+1} &= \hat{P}_{i+1|i}H_{i+1}^T S_{i+1}^{-1}.\end{aligned}$$

The denominator is given by:

$$p\Big(\tilde{\mathbf{y}}_{i+1}^{(k)} \mid \tilde{\mathbf{y}}_i^{(k)}, \ldots, \tilde{\mathbf{y}}_i^{(k)}\Big) = \begin{cases} p_y\Big(\mathbf{y}_{i+1}^{(k)}\Big)\, pd_{i+1}^{(k)} & \tilde{\mathbf{y}}_{i+1}^{(k)} \neq \emptyset \\ 1 - pd_{i+1}^{(k)} & \tilde{\mathbf{y}}_{i+1}^{(k)} = \emptyset \end{cases} \tag{2.22}$$

where:

$$p_y\Big(\mathbf{y}_{i+1}^{(k)}\Big) \sim N\Big(H_{i+1}\hat{m}_{i+1|i},\, H_{i+1}\hat{P}_{i+1|i}H_{i+1}^T + R_{i+1}\Big)\,.$$

Note that in this context, the distribution of the target's current state is:

$$p(\mathbf{x}_{i+1}^{(k)};\, \hat{m}_{i+1|i},\, \hat{P}_{i+1|i}),$$

which is used to calculate the parameter $pd_{i+1}^{(k)}$.

For the emission distribution implicitly defined by the joint Gaussian approximation for an arbitrary P_D model, the standard Kalman filter solution does not apply. This is because the numerator and integrand in (2.21) are not represented by the product of the emission and state distributions, as indicated here, but are instead replaced

by the approximation of the joint state and measurement distribution given by (2.17). In this case, the updated estimate is derived from the predicted state estimate as:

$$\hat{m}_{i+1|i+1} = \hat{m}_{i+1|i} + K_{i+1}\nu_{i+1}$$
$$\hat{P}_{i+1|i+1} = \hat{P}_{i+1|i} - K_{i+1}S_{i+1}K_{i+1}^T,$$

where:

$$\nu_{i+1} = \mathbf{y}_{i+1}^{(k)} - \left(\hat{\mu}_{i+1}^{(k)}\right)_y$$
$$S_{i+1} = \left(\hat{\Sigma}_{i+1}^{(k)}\right)_{yy} + R_{i+1}$$
$$K_{i+1} = \left(\hat{\Sigma}_{i+1}^{(k)}\right)_{xy} S_{i+1}^{-1}.$$

The notation $(\hat{\mu}_{i+1}^{(k)})_y$ is used to indicate the selection of the subvector corresponding to the measurement mean from the joint state and measurement mean vector, $\hat{\mu}_{i+1}^{(k)}$. The notation $(\hat{\Sigma}_{i+1}^{(k)})_{yy}$ is used to indicate the selection of the submatrix corresponding to the measurement covariance from the joint covariance matrix, $\hat{\Sigma}_{i+1}^{(k)}$. Similarly, $(\hat{\Sigma}_{i+1}^{(k)})_{xy}$ is used to represent the state-measurement cross-covariance.

Using $pd_{i+1}^{(k)}$ to denote the constant of proportionality, the denominator has the same form as before, but with:

$$p_y\left(\mathbf{y}_i^{(k)}\right) \sim N\left(\left(\hat{\mu}_i^{(k)}\right)_y, \left(\hat{\Sigma}_i^{(k)}\right)_{yy}\right).$$

Again, $p(\mathbf{x}_{i+1}^{(k)}; \hat{m}_{i+1|i}, \hat{P}_{i+1|i})$ is the distribution of the target's current state, which is used here when calculating the parameters $pd_{i+1}^{(k)}$, $\hat{\mu}_{i+1}^{(k)}$, and $\hat{\Sigma}_{i+1}^{(k)}$.

Therefore, to compute joint measurement probability, we must: run the Kalman filter; at each time step, use the predicted state estimate to define (2.22) based on the chosen model and evaluate it at the measured value; and take the product of these probabilities.

To complete the discussion of the target measurement likelihood, however, one final point remains to be made. That is, how to compute the numerator in (2.8). This is the joint distribution of the negative measurements, which can be obtained from the joint measurement distribution by integrating out the positive measurements or from the joint distribution of target states and measurements by integrating out both the states and the positive measurements. It is required to scale the joint measurement

probability in order to get the joint probability of the positive measurements, given the negative measurements, which is the target measurement likelihood.

The easiest way to compute the joint probability of the negative measurements is to used a version of the Kalman filter modified to skip the update steps to continue propagating the predictions. This works because integrating out a positive measurement is equivalent to skipping the update step in the corresponding scan during the Kalman filter process and simply taking the predicted state estimate to be the filtered state estimate. For negative measurements, the update does not influence the state for either definition of the emission distribution. Thus, to compute the joint negative measurement probability, we: run the modified Kalman filter; at each time step, we use the predicted state to estimate the discrete part of (2.22); and take the product of those probabilities.

2.3.1.1 Multiple Models

In the previous section, we assumed that the evolution of a target's state could be modeled as a discrete-time linear system driven by Gaussian noise so that the integrals in (2.10) can be solved exactly in linear time. This led to the system model given by equation (2.11) with the system matrix, F_j, control input, u_j, and process noise, w_j, which were all assumed to be known for any given time, j. However, for highly maneuverable targets, such those that we expect to encounter, there is often considerable uncertainty regarding those values.

One way to address this model uncertainty without abandoning the advantages of the linear-Gaussian framework is to assume that a target's motion can be described by one of a finite number of models, each in the form of (2.11). Thus, the motion of a target is assumed to be governed by the discrete mode in which it is operating and this unobserved mode can be used to specify the appropriate linear-Gaussian model for the state transition. The mode is assumed to be in effect throughout the sampling period. This gives us the following mode-conditional system model for the k-th target:

$$\mathbf{x}_j^{(k)} = F[M(j)]\,\mathbf{x}_{j-1}^{(k)} + u[M(j)] + w[M(j)]\,, \tag{2.23}$$

where $M(j)$ denotes the mode in effect during the j-th sample period. Note that the measurement model in (2.13) can be similarly extended to handle mode-conditional measurement matrices and noise, although we will not consider that here.

Having defined the state transitions as a function of the mode, the state of a

target at time j now depends on the sequence of modes in effect from the initial sample until that time. Let the l-th mode sequence through time j be denoted by $M^l(j)$. Since the sequence is unobserved, we can model it using a Bayesian framework as a random variable that is then integrated out. Thus, the marginal distribution of the state of the k-th target at time i becomes:

$$p\Big(\mathbf{x}_i^{(k)} \mid \tilde{\mathbf{y}}_i^{(k)}, \ldots, \tilde{\mathbf{y}}_1^{(k)}\Big) = \sum_{l=1}^{r^i} p\Big(\mathbf{x}_i^{(k)} \mid M^l(i), \tilde{\mathbf{y}}_i^{(k)}, \ldots, \tilde{\mathbf{y}}_1^{(k)}\Big)\, p\Big(M^l(i) \mid \tilde{\mathbf{y}}_i^{(k)}, \ldots, \tilde{\mathbf{y}}_1^{(k)}\Big), \tag{2.24}$$

where r is the number of modes. The probability of the mode sequence:

$$p(M^l(i) \mid \tilde{\mathbf{y}}_i^{(k)}, \ldots, \tilde{\mathbf{y}}_1^{(k)}),$$

is derived using Bayes' rule with the mode switching modeled as a Markov process with known transition probabilities.

The target state, given all the previous measurements, is thus modeled as a Gaussian mixture. To approximate the mixture using a single Gaussian so that (2.19) continues to hold, moment matching can be used. Unfortunately, the mixture contains an exponentially increasing number of terms so exact evaluation is not possible in practice. Instead, various suboptimal approximations are used to control the growth of the mode sequences. The most common such approaches are known as the generalized pseudo-Bayesian (GPB) and interacting multiple model (IMM) algorithms. These algorithms keep the last one (or two) step of the mode sequence and merge sequences that differ in their older steps. See [6] for a detailed discussion of these dynamic multiple model estimation algorithms. When the set of possible modes is not fixed, variable structure multiple model estimators can be used [115, 103].

Note that in some cases, rather than using a state estimate based on the weighted combination of the mode-conditional models, it may be advantageous to instead use the output from the model with the highest probability. This is straightforward using the GPB and IMM algorithms since they both compute the probability of the current mode, $M(i)$, conditional on the previous measurements, at each time step.

2.3.1.2 Nonlinear Models

For many real-world tracking applications, the assumption that the previous and current states are linearly related, as in (2.11), is too restrictive. Such models cannot

accurately describe target behavior such as path-constrained motion that is likely to be important. Similarly, the assumption that the current state and the corresponding measurement are linearly related, as in (2.13), is often too restrictive as well. An example of this from our motivating application is the nonlinear relationship between radar measurements, which are in polar coordinates, and the state, which is Cartesian coordinates.

In order to utilize nonlinear state transition and measurement models without losing the efficiency of the Gaussian representation, particularly the ability to solve the integrals in (2.10), we must find a Gaussian approximation to the joint distribution of the variables. One common approach is to replace the nonlinearity with a linear approximation, in which case the joint Gaussian distribution is easy to find. The well-known extended Kalman filter (EKF) does exactly that using an approximation based on the Taylor series expansion around the mean. Of course, this is effective only if the high order terms are negligible, which not usually the case in practice. It also requires that the nonlinearity is differentiable, which precludes discontinuous systems and limits its utility. Another approach is to directly approximate the distribution resulting from the nonlinear function. This is what the unscented transform (UT) does [91, 92].

Given two random variables, A and B, that are related through the nonlinear transformation $B = f(A)$, the UT can be used to estimate the moments of their joint distribution. In this case, our interest lies in the first two moments in order to form a Gaussian approximation. The UT operates by deterministically selecting a fixed number of so-called sigma points whose weighted sample moments match those of A. It is the weights that determine which moments are matched and to what degree. The sigma points are then propagated through the nonlinear function $f(\cdot)$ and the transformed values, along with the weights, are used to estimate the moments of the joint distribution. Thus, rather than attempting to approximate an arbitrary nonlinearity, the UT instead applies the nonlinear transformation directly and approximates the moments of the transformed values. Compared to using an approximation of the nonlinearity based on the Taylor expansion, the UT is easier to implement, more broadly applicable, and has proven to be more accurate.

Let $A \sim N(\mu_A, \Sigma_A)$ be Gaussian random variable in $\mathbb{R}^n$ that is related to the random variable $B \in \mathbb{R}^m$ as stated above. The UT provides an efficient means of

generating approximations of the form:

$$\begin{bmatrix} A \\ B \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix}, \begin{bmatrix} \Sigma_A & \Sigma_{AB} \\ \Sigma_{AB}^\top & \Sigma_B \end{bmatrix} \right).$$

The conditional distribution, $p(B \mid A)$, can be shown to follow a Gaussian distribution with the mean and covariance given by:

$$\begin{aligned} \mu_{B|A} =& \mu_B + \Sigma_{AB}^\top \Sigma_A^{-1} (A - \mu_A) \\ \Sigma_{B|A} =& \Sigma_B - \Sigma_{AB}^\top \Sigma_A^{-1} \Sigma_{AB}. \end{aligned}$$

The approximation based on the UT is generated as follows.

Let $\sigma_i \in \mathbb{R}^n$ be the i-th sigma point, where $1 \leq i \leq d$. Associated with each sigma point are a pair of scalar weights used to compute the unknown terms in the joint Gaussian approximation. The first weight in each pair is given by the elements of $W_m \in \mathbb{R}^d$ and used to compute the mean of B, which is:

$$\mu_B = \sum_{i=1}^{d} (W_m)_i \, f(\sigma_i).$$

The second weight in each pair is given by the elements of $W_m \in \mathbb{R}^d$ and used to compute the covariance terms:

$$\begin{aligned} \Sigma_B =& \sum_{i=1}^{d} (W_c)_i \, (f(\sigma_i) - \mu_B) \, (f(\sigma_i) - \mu_B)^\top \\ \Sigma_{AB} =& \sum_{i=1}^{d} (W_c)_i \, (\sigma_i - \mu_A) \, (\sigma_i - \mu_A)^\top. \end{aligned}$$

There is no one way to select the sigma points and the corresponding weights. The different methods reported in the literature differ in their accuracy, particularly with respect to the high order moments, and computational complexity. The sigma points and weights are often parameterized to allow for problem-specific tuning. In general, the sigma points are derived from the columns of $S = \sqrt{\Sigma_A}$, where $\Sigma_A = SS^\top$ and is computed using the Cholesky factorization. Together with the weights, they are selected by solving for the values that yield a discrete distribution with moments that match those of A. Note that the UT does not require that A follow a Gaussian distribution and different assumptions will lead to different matching requirements. Existing selection schemes include one for matching the first three moments of an arbitrary distribution

([93]) and the first four nonzero moments of a Gaussian distribution [94]. However, for Gaussian random variables we have found the exact monomials method from Lerner ([112]) to be the most effective. It leads to a unique specification of the sigma points and weights, without the need to find suitable parameters, and offers a way to trade off complexity for more accuracy through the choice of precision (3 or 5), which changes the number of sigma points.

Unfortunately, the UT does not guarantee that the covariance matrix resulting from the approximation is positive definite or even positive semi-definite. In practice, we must be aware of this and modify the matrix when necessary. Simply using L_2 regularization to drive the smallest eigenvalue to a small positive value is one possibility. Lerner ([112]) describes another approach in which the "closest" positive definite matrix is found by casting the problem as one of convex optimization.

2.3.2 Prior

The prior is composed of the product of the two distributions given by equations (2.2) and (2.3) in the hierarchical model. It is the joint distribution of the indicator variables, Z and $\mathcal{Z}0$, that collectively represent an interpretation of the observed data. This distribution is also conditional on two parameters whose values are assumed to be unknown. These parameters are introduced into the model as random variables with hyperpriors given by the remaining two equations in the hierarchical model, (2.1) and (2.4). They will be discussed here along with the prior.

As noted in the preliminaries, the information contained in $\mathcal{Z}0$ is in fact redundant. Any false alarm indicator in $\mathcal{Z}0$ can be readily derived from the corresponding cluster label in Z as:

$$z0_i = \begin{cases} 0 & z_i > 0 \\ 1 & z_i = 0, \end{cases}$$

for $1 \leq i \leq nm$.

The reason for introducing these additional indicator variables is to allow us to separate the treatment of the false alarm cluster from that of the target clusters, in the prior. Although the false alarm cluster could be thought of as just another cluster, our prior beliefs about it are sufficiently different from those regarding the target clusters to a warrant such treatment in the interest of clarity. In contrast to the appearance of targets in the surveillance region, which is generally regarded as unpredictable, we

assume that the rate at which false alarm measurements appear in each scan can be well modeled. This implies some relatively strong beliefs about the size of the false alarm cluster that should be reflected in the prior. At the same time, we would like to remain agnostic about the size of any given target cluster. Most of the types of priors used for clustering models make it difficult, if not impossible, to describe such asymmetric prior knowledge in a truly unified manner. This is indeed the case for all but one of the models that we will consider in the next chapter. An important consequence of modeling the false alarm indicators separately is that is allows us to exploit the relationship between the number of false alarm measurements and the number of target clusters to define the hyperprior given by (2.4).

To separately model our prior beliefs about the false alarm cluster labels and the target cluster labels, while still enforcing the necessary dependencies between Z and $\mathcal{Z}0$, we have factored the prior as:

$$p(Z,\, \mathcal{Z}0 \mid \alpha,\, \lambda) = p(Z \mid \mathcal{Z}0,\, \alpha)\, p(\mathcal{Z}0 \mid \lambda)\,.$$

The distribution $p(\mathcal{Z}0 \mid \lambda)$ represents the prior over the false alarm cluster labels, while the distribution $p(Z \mid \mathcal{Z}0,\, \alpha)$ is said to represent the prior over the target cluster labels; the latter requires some qualification. Conditional on $\mathcal{Z}0$, the variables in Z can be partitioned into two independent subsets so that the first component of the prior becomes:

$$p(Z \mid \mathcal{Z}0,\, \alpha) = p(Z_{\tau_0} \mid \mathcal{Z}0,\, \alpha)\, p(Z_{\tau_+} \mid \mathcal{Z}0,\, \alpha)\,,$$

where Z_{τ_0} contains those variables associated with false alarm measurements and Z_{τ_+} contains those associated with target measurements. The joint distribution of the variables associated with false alarm measurements is defined simply as:

$$p(Z_{\tau_0} \mid \mathcal{Z}0,\, \alpha) = \begin{cases} 1 & \text{if } z_i = 0 \text{ for } z_i \in Z_{\tau_0} \\ 0 & \text{otherwise} \end{cases}$$

and the joint distribution of the variables associated with target measurements is subject to the constraint that $p(Z_{\tau_+} \mid \mathcal{Z}0,\, \alpha) = 0$ if $z_i = 0$ for $z_i \in Z_{\tau_+}$. This ensures that the indicator values remain consistent between Z and $\mathcal{Z}0$, and that the two components of the prior each address the appropriate cluster type. The joint distribution of the variables associated with target measurements is really what is meant when we refer

to the prior distribution of the target cluster labels. However, for convenience we will continue to call $p(Z \mid \mathcal{Z}0, \alpha)$ the prior distribution of the target cluster labels with the understanding that its definition is based on the appropriate subset of Z.

2.3.2.1 Distribution of the False Alarm Cluster Labels

The probability mass function of the prior distribution over the false alarm cluster labels is denoted by $f_{\mathcal{Z}0}(\mathcal{Z}0; \lambda)$ to indicate that it does not match any well-known distribution. It contains a single parameter, λ, that is the false alarm rate in units of false alarms per scan.

To define this component of the prior, we begin by considering which of the information encoded in $\mathcal{Z}0$ will be used. The set of false alarm indicators tells us, among other things, the number of measurements in each scan deemed to be false alarms. It is based on these counts that the probability mass function is defined. Using λ, we can express the probability of a given number of false alarms occurring in a single scan. As a function of $\mathcal{Z}0$, the upper limit of the support of this distribution over the false alarm counts must be restricted to the total number of measurements received during the scan to ensure that it sums to 1. Therefore, the distribution over false alarm counts must be conditioned on the number of measurements in the scan. Assuming that false alarm measurements are independent across scans, the joint distribution of the false alarm indicator variables, given the false alarm rate, can be written as:

$$p(\mathcal{Z}0 \mid \lambda) = \prod_{i=1}^{ns} p(nfa_{s_i} \mid \lambda,\, nm_{s_i}) \,,$$

where nfa_{s_i} is the number of false alarms and nm_{s_i} is the number of measurements, in scan i. Note that nm_{s_i} appears in the conditioning without any corresponding marginal probability because it is deterministic knowledge.

The number of false alarms in scan i, nfa_{s_i}, is a discrete random variable assumed to follow a Poisson distribution with rate parameter λ and truncated so that the support is $[0..nm_{s_i}]$. Therefore, each product term is given by:

$$\begin{aligned} p(nfa_{s_i} \mid \lambda,\, nm_{s_i}) &= \frac{\lambda^{nfa_{s_i}} \exp(-\lambda)}{(nfa_{s_i})!} \frac{(nm_{s_i})!}{\Gamma(nm_{s_i}+1,\, \lambda)} \\ &= \frac{\lambda^{nfa_{s_i}}}{(nfa_{s_i})! \left(\sum_{a=0}^{nm_{s_i}} \frac{\lambda^a}{a!}\right)} . \end{aligned}$$

Although its exact value is assumed to be unknown, the false alarm rate is assumed to be something that we can characterize probabilistically, either empirically or based on theoretical models. Typically, this done empirically in order to capture the unmodeled effects of the environment on the detector. For simplicity, we use a gamma distribution, as indicated in (2.1), to describe the uncertainty in the false alarm rate since this is the conjugate prior for the Poisson rate parameter. Thus, we have:

$$p\left(\lambda \mid a_\lambda, b_\lambda\right) \sim \Gamma\left(a_\lambda, b_\lambda\right),$$

where a_λ is the shape and b_λ is the rate. This enables the associative tracking model to deal with variations in the false alarm rate and the extra variance that that introduces into the false alarm count distributions. However, when the false alarm rate is found to vary systematically as a function of the scan (i.e. time) then modeling it as a non-homogeneous Poisson process would probably be more effective [125], but we will not consider that situation here.

The parameters of the gamma hyperprior are assumed to be known. Suitable values can be obtained using the parametric empirical Bayes method [25] described as follows. Consider the number of false alarms in a given scan as a Poisson random variable without regard to *Z0*. Its distribution is given by:

$$p(nfa_{s_i} \mid \lambda) = \frac{\lambda^{nfa_{s_i}} \exp\left(-\lambda\right)}{\left(nfa_{s_i}\right)!}.$$

The rate parameter is itself a gamma random variable, so its distribution is given by:

$$p(\lambda \mid a, b) = \frac{b^a}{\Gamma\left(a\right)} \lambda^{(a-1)} \exp\left(-b\lambda\right),$$

where subscripts on the parameters have been dropped to avoid confusion. All the information about the parameters is contained in the marginal distribution of the false alarm counts:

$$\begin{aligned} p(nfa_{s_i} \mid a, b) &= \int p(nfa_{s_i} \mid \lambda)\, p(\lambda \mid a, b)\, d\lambda, \\ &= \left(\begin{array}{c} nfa_{s_i} + a - 1 \\ a - 1 \end{array} \right) \left(\frac{b}{b+1}\right)^a \left(\frac{1}{b+1}\right)^{nfa_{s_i}}. \end{aligned}$$

This can be recognized as a negative binomial distribution with parameters: a and $p = \frac{b}{b+1}$. Thus, the parameters of the gamma hyperprior can be derived by fitting a

negative binomial distribution to empirical false alarm count data. Let $\hat{r}$ and $\hat{p}$ be the estimated parameters of the negative binomial distribution. They can be computed using any preferred method (e.g. see [172] and the references therein). Unfortunately, there exists a closed form solution for the maximum likelihood estimate of just one of these parameters. The parameters of the gamma hyperprior are then given by: $a_\lambda = \hat{r}$ and $b_\lambda = \frac{\hat{p}}{1-\hat{p}}$.

Here it arose as a mixture of Poisson distributions with Gamma mixing weights in the course of a parameter estimation procedure, but the negative binomial distribution is itself a perfectly valid model for the number of false alarms in a given scan. In fact, it can be viewed as a generalization of the Poisson distribution. Our decision to use a Poisson distribution to describe the false alarm counts was somewhat arbitrary, but motivated by the fact that this is the standard approach for tracking models. We then extended that approach by modeling the Poisson rate parameter as a random variable, to be included in the inference process, based on the beliefs that knowing the exact rate and the perfect constancy of the rate are unlikely to satisfied in practice. When the assumptions underlying the Poisson distribution are not well satisfied, once consequence is that the false alarm counts can be overdispersed with respect to that model. In other words, the true variance is larger than what the Poisson distribution prescribes, which is that the variance is equal to the mean. As a model for the distribution of counts, the negative binomial has more freedom to describe such behavior. One could opt to replace equations (2.2) and (2.3) in the hierarchical model with $p(\mathcal{Z}0)$ defined based on the product of truncated negative binomial distributions. However, further consideration of this alternative will be left to future work.

2.3.2.2 Distribution of the Target Cluster Labels

The probability mass function of the prior distribution over the target cluster labels is denoted by $f_Z(Z;\, \mathcal{Z}0,\, \alpha)$ to indicate that it does not necessarily match any well-known distribution. In some cases it may indeed match a well-known distribution, so this is really more of a place holder because we will be looking in detail at various ways to define this distribution in the next chapter. The best way to represent our prior beliefs about the target cluster labels is an open question in need of further study.

Defining this component of the prior is particularly challenging because we wish to remain largely agnostic about the number of targets and the sizes of the target clus-

ters, which are all assumed to be a priori unknown. This implies that any assumptions along these lines that we do introduce should be as weak as possible. One promising approach is to use data-driven nonparametric priors. We will defer to the next chapter the definition of such priors. At present, let it suffice to say that an important class of these nonparametric priors has a single so-called concentration parameter, $\alpha \in \mathbb{R}^+$, that controls the formation of new clusters. Since the number of clusters is assumed to be unknown, it follows that α too is unknown. The remainder of this section will be devoted to outlining how the hyperprior for α is defined.

We begin by noting that the distribution of α has been defined, in equation (2.4), as conditional on $\mathcal{Z}0$. This is because the concentration parameter controls the expected number of clusters for a given number of variables and $\mathcal{Z}0$ tells us not only the number of measurements deemed to be false alarms but also the number deemed to be targets. In the context of this component of the prior, the number of measurements deemed to be targets is the number of variables to be clustered. Let this number be denoted by nm_{τ_+}.

In the absence of any other constraints, the expected number of clusters approaches one as α approaches zero. As α approaches infinity, the expected number of clusters approaches the number of variables, nm_{τ_+}. The concentration parameter thus acts like a control knob for the distribution of the number of clusters implied by $p(Z \mid \mathcal{Z}0, \alpha)$, adjusting the distribution so that its expected value moves between those two extremes. This general behavior is true of all the nonparametric priors that we will consider. It is the rate at which the expected value changes and the shape of the distribution that varies across models. Of course, the different priors will also differ substantially in ways that are not a function of α such as with respect to the distribution over the sizes of the clusters. Our preferences regarding these characteristics influences our choice of the prior itself, but not the hyperprior.

There is, however, an important constraint that does impact the expected number of clusters. It follows from the earlier assumption that there are no split measurements. If there are no split measurements, then each target measurement in a given scan must have originated from a different target and hence be assigned to a different cluster. We will refer to this requirement that the target measurements within a scan have unique cluster labels as the scan constraint. One consequence of this constraint is that it places a lower bound on the number of clusters because that number cannot

be less the maximum scan depth (i.e. the maximum number of target measurements in any scan) when the scan constraint is satisfied for all of Z. Let that lower bound be denoted by $K_{min} = maxDepth$, whose value can be obtained from $\mathcal{Z}0$ since the set of false alarm indicator variables effectively encodes the scan depths.

Note that in their default forms, the nonparametric priors do not reflect this scan constraint so they will have to be modified. Here, we will assume that the prior has been appropriately modified so that α predicts the correct distribution of the number of clusters. The scan-constrained forms of the priors will be discussed in the next chapter.

The distribution $p(\alpha \mid \mathcal{Z}0)$ should be defined so that the most density coincides with the values of α whose implied distribution of the number of clusters is the most reasonable in the sense that it conforms to our prior beliefs. Let us now contemplate what assumptions we might be willing to make regarding the distribution of the number of clusters, ignoring for the moment those implied by our choice of $p(Z \mid \mathcal{Z}0, \alpha)$. We begin by recalling our desire to remain agnostic about the number of targets. Does this then imply that all numbers of clusters should be regarded as equally likely? That is hardly a reasonable presumption for a tracking application; especially not for a surveillance application. The extreme case of nm_{τ_+} clusters should be regarded as rather unlikely. A more reasonable belief would be that having a small number of clusters is more probable than a large one. This suggests a template for describing our beliefs about the number of clusters: a distribution that decreases as a function of the number of clusters. It does not entail any strong assumptions about specific values but instead expresses a general preference for a relatively small number of clusters. Let the distribution of the number of clusters that represents these beliefs be denoted by $p_0(K \mid \mathcal{Z}0)$, where K is the number of clusters.

To define $p_0(K \mid \mathcal{Z}0)$ in a way that scales with the number of variables, we introduce two new parameters. The first is *minClusterSz*. As the name suggests, it describes the minimum number of variables that constitute a cluster. It is not meant to be a hard limit but rather a means to determine a more realistic upper bound on the number of clusters for this distribution. Let that upper bound be denoted by K_{max}, whose value is the maximum number of clusters, consistent with the scan constraint, computed based on *minClusterSz* and the scan depths implied by $\mathcal{Z}0$. In practice, we use:

$$K_{max} = \max\left(K_{min}, \left\lceil \frac{nm_{\tau_+}}{minClusterSz} \right\rceil\right).$$

The motivation for using an upper bound other than the number of variables is that singleton clusters are not meaningful in a tracking application. One measurement does not provide enough information to identify it as a target rather than a false alarm. Therefore, we have $minClusterSz \geq 2$. Values greater than two should be interpreted as an expression of belief about the expected minimum cluster size. The practical effect of increasing $minClusterSz$ is to compress $p_0(\cdot)$ and intensify the preference for a small number of clusters. The second variable is η, which describes the ratio:

$$\frac{p_0(K = K_{max} \mid Z0)}{p_0(K = K_{min} \mid Z0)}.$$

It provides a convenient means of parameterizing the decay function used to define the distribution by relating the probabilities of the least probable number of clusters and the most probable number. The distribution of the number of clusters that represents our prior beliefs is then given by:

$$p_0(K \mid Z0) = \frac{a(1-a)^K}{(1-a)^{K_{min}} - (1-a)^{1+K_{max}}}, \tag{2.25}$$

where $a = 1 - \eta^{\frac{1}{K_{max} - K_{min}}}$ and $K \in [K_{min}..K_{max}]$. It is proportional to a geometric distribution, which yields the following interpretation. We begin with K_{min} clusters and conduct a series of simple random experiments to determine how many to add. With probability $1-a$ we add a new cluster and with probability a we keep the current number of clusters. The sequence of experiments stops when we encounter the first decision to keep the current number of clusters. The probability of stopping with K clusters is proportional to $p_0(K \mid Z0)$. Figure 2.3 shows an example of the prior beliefs about the number of clusters given 25 measurements, with 5 measurements made in each scan. Although derived differently, the resulting distribution resembles the prior for the number of clusters proposed by Casella et al. [26]. They assumed that the number of clusters followed a truncated Poisson distribution with an unknown rate parameter, described by the intrinsic prior, and ended up with a similar exponential-like decay form.

We can now define $p(\alpha \mid Z0)$ by relating our beliefs about the number of clusters, as expressed by $p_0(K \mid Z0)$, to the distribution of the number of clusters that is implied by the prior over the target cluster labels, $p(Z \mid Z0, \alpha)$. Let the implied distribution of the number of clusters be denoted by $p_Z(K \mid Z0, \alpha)$. The two cluster count

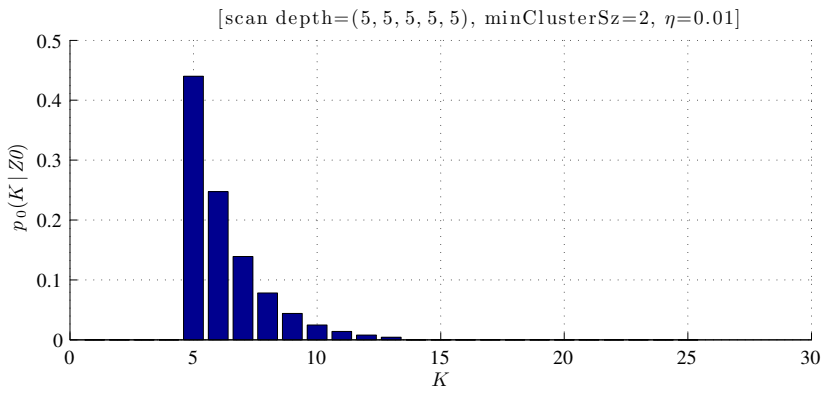


Figure 2.3: Example of prior beliefs about the number of clusters based on (2.25).

distributions are related using their Jensen-Shannon divergence to quantify their similarity as a function of α. The Jensen-Shannon divergence (JSD) between $p_Z(K \mid Z0, \alpha)$ and $p_0(K \mid Z0)$ is defined by:

$$JSD\left(p_Z \parallel p_0\right) = \frac{1}{2} D\left(p_Z \parallel M\right) + \frac{1}{2} D\left(p_0 \parallel M\right),$$

where $M = \frac{1}{2}\left(p_z + p_0\right)$ and $D\left(\cdot\right)$ is the Kullback-Leibler (KL) divergence, which is defined for discrete probability distributions P and Q to be:

$$D\left(P \parallel Q\right) = \sum_i P\left(i\right) \ln\left[\frac{P\left(i\right)}{Q\left(i\right)}\right].$$

The JSD is computed for $K \in \left[K_{min}..nm_{\tau_+}\right]$ because the upper bound of K_{max} only applies to $p_0(K \mid Z0)$ so we use the larger of the two. To convert this into a probability distribution, we shift and scale the negative JSD to get:

$$p\left(\alpha \mid Z0\right) = \frac{-JSD\left(p_Z \parallel p_0\right) + D_0}{C}, \tag{2.26}$$

where D_0 is a constant needed to make the values non-zero over the support and C is the normalizing constant. The support is defined to be the interval $(0, \alpha_{max}]$, where α_{max} is the value of the concentration parameter for which the expected number of clusters implied by $p(Z \mid Z0, \alpha)$ is K_{max}. The constant D_0 is given by the $JSD\left(p_Z \parallel p_0\right)$ evaluated at α_{max}. It turns out that this distribution can be well approximated by a gamma distribution, hence the definition:

$$p\left(\alpha \mid Z0\right) \sim \Gamma\left(a_\alpha(Z0), b_\alpha(Z0)\right).$$

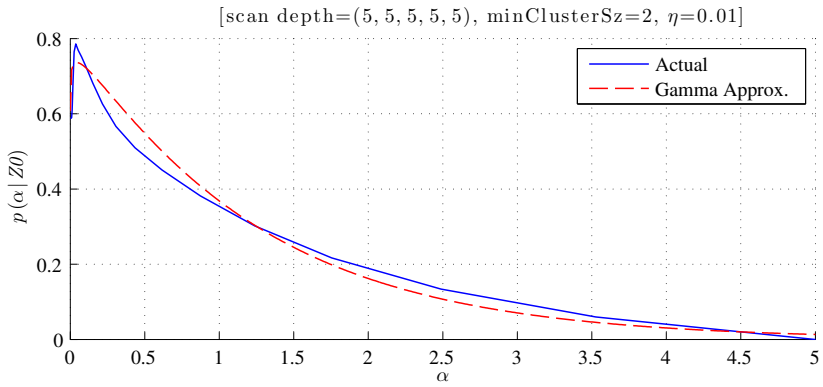


Figure 2.4: Example of the hyperprior for the concentration parameter of a scan-constrained Dirichlet process.

The parameters are functions of $\mathcal{Z}0$ and computed by fitting a gamma distribution to (2.26). In practice, this can be done efficiently using one of the moment and mode matching methods described in appendix A. The moments can be estimated by evaluating $-JSD\left(p_Z \parallel p_0\right) + D_0$ at small number of sample points and then computing the mean and variance using the trapezoidal rule to approximate the sample weights. If necessary, the mode can be computed using a derivative-free optimization method such as the Brent algorithm. Figure 2.4 shows an example of $p(\alpha \mid \mathcal{Z}0)$, including both (2.26) and its gamma-based approximation, for a scan-constrained Dirichlet process prior.

2.4 Inference

Inferences about what is happening within the surveillance region are made using a sample drawn from the joint posterior, $p(Z, \mathcal{Z}0, \lambda, \alpha \mid \tilde{Y}, a_\lambda, b_\lambda)$. From such a sample, we can obtain Monte Carlo estimates of all posterior quantities of interest in the form of expectations of arbitrary functions of the indicator variables, false alarm rate, and concentration parameter (when applicable). The choice of these functions will depend on the application and the quality of the estimates will depend on how the sample was obtained.

2.4.1 Markov Chain Monte Carlo (MCMC)

Being unable to sample from the joint posterior directly, we can use Markov Chain Monte Carlo (MCMC) methods to simulate draws from this distribution. MCMC

methods are a family of algorithms for generating a sample from an arbitrary distribution on a space Ω by constructing a Markov chain with states $\omega \in \Omega$ that has the desired distribution as its stationary distribution, which is denoted by $\pi(\omega)$. The sample then consists of the states visited by the chain after converging to its equilibrium. In order to ensure that the chain does in fact converge and to the desired distribution, it must satisfy certain conditions. Specifically, the chain must be: irreducible, aperiodic, and positive recurrent. Irreducible means that there is a positive probability of the chain reaching a given state from any other state, although not necessarily in one step. Aperiodic means that the chain does not return to the same state at some fixed interval (i.e. it does not get trapped in a cycle). Finally, positive recurrent means that for any given state, the chain will eventually return to that state with probability 1. A chain that satisfies all of these conditions is said to be ergodic, which implies that $\pi(\omega)$ exists and is unique. The ergodic theorem guarantees that given such a chain, the Monte Carlo estimates of the expectation of any function with respect to $\pi(\omega)$, computed using sample means, will converge almost surely to their to true values despite the serial dependence between the sample points due to the Markov property [163]. However, the fact that the sample points are not independent does significantly complicate the calculation of the variability of these estimates.

It is relatively easy to construct an ergodic chain with $\pi(\omega)$ equal to the posterior distribution so that estimates of the posterior expectations are theoretically guaranteed to converge. However, there is no general, practical, and reliable way to determine how long the chain must be run to produce an estimate with the desired level of accuracy. A substantial amount of research has been dedicated to determining convergence rates and stopping rules to ensure that the sample is large enough to adequately represent the underlying stationary distribution [36, 19]. Unfortunatey, none of these methods for diagnosing convergence are universally appropriate or accepted. The most rigorous methods are not widely applicable and can be overly pessimistic. The most commonly used methods are graphical and require a subjective assessment of their output.

In principle, the problem of deciding when to stop the chain can be avoided by using so-called perfect sampling algorithms, such as Propp-Wilson [152, 163], which are guaranteed to produce i.i.d. samples from the stationary distribution of the Markov chain. In practice, however, there are a number of problems that severely limit their usefulness. Chief among these problems is the running time. It is difficult to determine

in advance how long it will take a perfect sampling algorithm to produce the requested sample size. Thus, we often do not know whether a sample large enough to achieve the desired accuracy can be obtained given a fixed time budget. Of course, the independence of the sample points that we do obtain guarantees that we can easily compute the Monte Carlo variance of the estimates to determine how close we are to the desired accuracy. When knowing the variance is not critical, more accurate results can be obtained by simply running the MCMC sampler for the same amount of time [66]. The running time for perfect sampling algorithms depends on how fast the chain converges to its stationary distribution, which is the same fundamental performance bottleneck impacting traditional MCMC sampling. This means that when the chain is slow to converge, we need either a large MCMC sample or have to wait a long time for the perfect samples. Thus, perfect sampling is not interesting because it is efficient, it is not, but rather because it frees us from having to monitor the output of the sampler. In any case, perfect sampling is currently only an option for problems in which the state space satisfies certain theoretical conditions that can be exploited to devise a practical algorithm. Those conditions do not apply here.

The reality is that we can never be certain about how well our estimates have converged and thus how accurate they are. This is especially true when the state space is very high-dimensional, as is the case for the associative tracking model. Dutifully applied MCMC convergence diagnostics can at best be a rough guide; at worst they can be misleading. However, we should not be discouraged by the circumstances; just careful about how we interpret the output. We can still construct a sampler that works well enough in practice. MCMC methods have yielded useful solutions to similar high-dimensional problems, such as image segmentation [7, 186] and target tracking [135, 60, 201], without establishing convergence. For such problems, a qualitative assessment of the output given known inputs is actually quite revealing. Although that cannot quantify the variance in the estimates, nor can it quantify how close the sample distribution is to the stationary distribution, it can give us confidence that the chain has explored all the important features of the distribution such as the modes. This is true because these particular problems are essentially labeling problems and for known inputs, we know a lot about the structure of the output, including what the what the MAP estimate should look like.

As Robert and Casella [163] point out, in light of the asymptotic validity of

MCMC, establishing convergence to the stationary distribution is not the most important practical concern but rather maximizing the speed with which the chain explores the support of the distribution. Accurate estimates require a sample that reflects the features of the underlying distribution and this criteria is most likely to be met for a given sample size when the chain explores the support of the distribution quickly. They also cite the correlation between sample points as an issue of practical concern. But this too is related to the speed of exploration as a highly correlated sample is symptomatic of the chain spending too much time in some regions before moving on. However, unless the correlation is extremely high, it will probably not significantly impact the precision of the estimates because the correlation only indicates that sample points generated close in time are unrepresentative and that impact will tend to be averaged out [117]. Some degree of correlation will always be present, however, because of the Markov dependence between successive sample points. As is the case for the convergence diagnostics, analyzing the correlation does not provide any firm guidance.

How fast the chain explores the support of the stationary distribution is often referred to as the mixing speed of the chain. The mixing time of a Markov chain is defined as the time (i.e. number of iterations) required for the distance to its stationary distribution to be small in some formally defined sense, usually based on variation distance. A chain that mixes rapidly will explore the support of the stationary distribution quickly and exhibit weak dependence on the initial conditions. Estimates of the expectations computed using a sample from such a chain will converge quickly, which is exactly what we want. However, before we consider what design decisions can speed up the mixing, let us first look at how to construct an ergodic chain with the desired stationary distribution.

The most general MCMC method is the Metropolis-Hastings algorithm [82], which works as follows. Let $\pi(\omega)$ be the posterior distribution that we wish to simulate and $q(\omega' \mid \omega)$ be the proposal distribution, which describes the conditional probability of moving to state ω' from state ω. The posterior distribution need only be known up to the normalizing constant. The proposal distribution must be something that we can readily evaluate in its fully normalized form and sample from. The algorithm begins with an initial state chosen at random. At the t-th iteration, a candidate state, $\tilde{\omega}$, is

generated from $q(\tilde{\omega} \mid \omega^{(t-1)})$. With probability:

$$a\Big(\tilde{\omega},\, \omega^{(t-1)}\Big) = \min\left\{1,\, \frac{\pi\Big(\tilde{\omega}\Big)\, q\Big(\omega^{(t-1)} \mid \tilde{\omega}\Big)}{\pi\big(\omega^{(t-1)}\big)\, q\big(\tilde{\omega} \mid \omega^{(t-1)}\big)}\right\}, \tag{2.27}$$

the proposal is accepted and the chain moves to the candidate state so $\omega^{(t)} = \tilde{\omega}$; otherwise the proposal is rejected and the chain remains in the same state so $\omega^{(t)} = \omega^{(t-1)}$. This process is repeated for the desired number of iterations.

The particular form of the acceptance probability ensures that the stationary distribution of the chain exists and is equal to $\pi(\omega)$. Note that it requires each state transition to be reversible. To ensure that the chain is ergodic, which guarantees the uniqueness of the stationary distribution and the desired convergence properties, the proposal distribution must satisfy a couple of mild conditions. Namely, $q(\cdot)$ must be defined such that there is nonzero probability that the chain remains in the same state from one iteration to the next and all states are reachable in a finite number of steps.

Green showed that the Metropolis-Hastings algorithm could be generalized in a variety of ways [73] . The one of interest here is the state-dependent mixing of Markov chain transition kernels. This allows us to consider a family of proposal distributions, $q_i(\omega' \mid \omega)$, where $i \in \mathcal{M}$, with weights $c_i(\omega)$ that depend on the current state, such that $c_i(\omega) \geq 0$ and $\sum_i c_i(\omega) \leq 1$. In this case, at each iteration of the algorithm a random proposal index i is chosen with probability $c_i(\omega^{(t-1)})$ and the candidate state is then generated from $q_i(\tilde{\omega} \mid \omega^{(t-1)})$. The proposal is accepted with probability:

$$a\Big(\tilde{\omega},\, \omega^{(t-1)}\Big) = \min\left\{1,\, \frac{\pi\Big(\tilde{\omega}\Big)\, c_i(\tilde{\omega})\, q_i\Big(\omega^{(t-1)} \mid \tilde{\omega}\Big)}{\pi\big(\omega^{(t-1)}\big)\, c_i\big(\omega^{(t-1)}\big)\, q_i\big(\tilde{\omega} \mid \omega^{(t-1)}\big)}\right\}. \tag{2.28}$$

With the remaining probability, $1 - \sum_i c_i(\omega^{(t-1)})$, no candidate is generated and the chain remains in the current state. The transitions of the chain are determined by the kernel corresponding to the mixture proposal $q_I(\tilde{\omega} \mid \omega^{(t-1)}) + \sum_i c_i(\omega^{(t-1)}) q_i(\tilde{\omega} \mid \omega^{(t-1)})$, where q_I is the identity proposal which always returns $\omega^{(t-1)}$.

Given a multivariate state, $\omega = (\omega_1, \omega_2, \ldots, \omega_n)$, and a specific choice of proposal distribution at each iteration, the Metropolis-Hastings algorithm is equivalent to another MCMC method known as the Gibbs sampler [63]. Gibbs sampling works by repeatedly sweeping over each variable, updating the corresponding part of the state with a new value generated from the distribution of that variable conditioned on all other variables. Each iteration of the algorithm corresponds to a particular sweep and

a particular variable within that sweep. At the t-th iteration, assume that variable i is being updated. A new value for that variable, $\tilde{\omega}_i$, is generated from $p(\tilde{\omega}_i \mid \omega_{\setminus i}^{(t-1)})$, where $\omega_{\setminus i}^{(t-1)}$ represents the values of all variables other than i from the previous iteration. This is equivalent to the candidate state $(\tilde{\omega}_i, \omega_{\setminus i}^{(t-1)})$ drawn from the proposal distribution $q(\tilde{\omega}_i, \omega_{\setminus i}^{(t-1)} \mid \omega^{(t-1)})$. It can be shown that the Metropolis-Hastings acceptance probability in this case is always 1, so $\omega^{(t)} = (\tilde{\omega}_i, \omega_{\setminus i}^{(t-1)})$. The variables are updated in some predetermined, or random, order and all variables are updated before proceeding to the next sweep. This process is repeated for the desired number of iterations. Note also that the sweeps and corresponding updates can be performed over blocks of variables as well. The proposal distribution $q(\tilde{\omega} \mid \omega^{(t-1)})$ is equal to the product of the proposal distributions used to update each variable (or block of variables), $\prod_{i=1}^{n} q(\tilde{\omega}_i, \omega_{\setminus i}^{(t-1)} \mid \omega^{(t-1)})$.

Gibbs sampling eliminates the often difficult task of selecting an appropriate proposal distribution but requires instead the ability to sample from the full conditional distribution of each variable (or block of variables). In some cases, the conditional distributions have well-known forms and can be easily sampled from. Otherwise, a hybrid approach commonly referred to as Metropolis-within-Gibbs, but more appropriately known as componentwise Metropolis-Hastings, can be employed. It utilizes a single iteration of the Metropolis-Hastings algorithm to update a variable (or block of variables) within the Gibbs sampler paradigm. In this case, the advantage of continuing to work within the Gibbs paradigm framework is that it simplifies complex high-dimensional problems by breaking them down into a sequence of simple low-dimensional problems. Of course, even when we can sample directly from the full conditionals, the Gibbs sampler does not necessarily produce a chain that mixes quickly. Indeed, a well-known limitation of Gibbs sampling is that it can be slow to mix when the state space is high-dimensional and there are strong couplings between variables that are not sampled together [7]. In this situation, we must reparameterize the problem so that the strongly coupled variables are placed in the same block. However, this often leaves us with conditional distribution that we cannot sample from directly and the only way forward is to try to find an effective proposal distribution to use with a Metropolis-Hastings update.

2.4.2 Population Monte Carlo (PMC)

Of course, MCMC is not the only approach to sampling from an arbitrary distribution such as our joint posterior. Importance sampling is another Monte Carlo method that can be used; one that offers certain advantages over MCMC. It is a family of algorithms based on the idea that expectations with respect to any arbitrary distribution, $\pi(\omega)$, can be represented as:

$$\mathbb{E}\left[g(\omega)\right] = \int g(\omega)\,\frac{\pi(\omega)}{q(\omega)} q(\omega)\,d\omega,$$

where $g(\omega)$ is some function of interest of the state and $q(\omega)$ is a proposal distribution, known as the importance distribution, from which sampling is easy. Given a sample of size N drawn from the importance distribution, we can then compute an unbiased estimate of the expectation as:

$$\mathbb{E}\left[g(\omega)\right] \approx \frac{1}{N}\sum_{i=1}^{N} g(\omega_i)\,\frac{\pi(\omega_i)}{q(\omega_i)},$$

where the $\frac{\pi(\omega_i)}{q(\omega_i)}$ are known as the importance weights. The variance of the estimate depends on the choice of importance distribution, but unlike for estimates based on an MCMC sample, it can be readily computed because the samples are independent. Another advantage over MCMC is that importance sampling does not require any convergence diagnostics or stopping rules because it always yields a valid approximation.

There are, however, a couple of difficulties with this estimator in practice. First, it requires $\pi(\omega)$ to be a properly normalized distribution, which is often not the case. Finally, for some choices of $q(\omega)$, the ratio $\frac{\pi(\omega)}{q(\omega)}$ is unbounded which will lead to infinite variance in the estimate. In general, the conditions sufficient to avoid this fate are difficult to guarantee [163]. Both of these issues can be circumvented by using the self-normalized importance sampling estimator given by:

$$\mathbb{E}\left[g(\omega)\right] \approx \frac{\sum_{i=1}^{N} g(\omega_i)\,\frac{\pi(\omega_i)}{q(\omega_i)}}{\sum_{i=1}^{N} \frac{\pi(\omega_i)}{q(\omega_i)}},$$

which normalizes the importance weights. This estimator has finite variance and is guaranteed to converge almost surely as $N \to \infty$. It does, however, introduce a small bias. Despite that fact, the stability of the self-normalized estimator makes it the preferable form in most cases.

Importance sampling can be an attractive alternative to MCMC because it is often a more efficient approach to simulation of the target distribution. It can also work where MCMC algorithms have trouble converging, even when using the same proposal distribution. As with MCMC, the performance of importance sampling strongly depends on the choice of proposal distribution. When discussing MCMC, we considered random walk proposals of the form $q(\omega' \mid \omega)$, which depend on the current state. In contrast, here we have assumed that independent proposals of the form $q(\omega)$ are available. For high-dimensional problems, these are considerably more difficult to construct than dependent proposals. In this case, population Monte Carlo (PMC) [24] provides a framework that allows us to enjoy the benefits of importance sampling while taking advantage of dependent proposal distributions constructed for MCMC.

PMC is a form of iterated importance sampling. It works by simulating a finite sample from $\pi(\omega)$ as opposed to generating a series of individual sample points using an approximation of the target distribution, as is the case with MCMC. The algorithm begins by producing an initial sample of size n by any convenient means. At each iteration, every sample point is then replaced with a new one generated from the proposal distribution, conditional on its previous value. The importance weights are computed for each sample and then the population of sample points is corrected by resampling them, with replacement, based on their weights. This resampling step avoids keeping sample points with negligible weights, thus avoiding the problem of degeneracy. It also ensures that the corrected sample will yield a convergent estimate after every iteration.

An important feature of PMC is that the proposal distribution can also be adapted at each iteration based on the performance of earlier proposals. Since the algorithm does not depend on approximating $\pi(\omega)$ using an ergodic Markov chain, these adaptations can be made with almost no constraint on the form of dependence on the past because there is no concern about destroying the ergodicity. For a suitable adaptation scheme, the variance of the estimates will decrease with increased iterations, although it cannot be driven to zero when the self-normalizing importance sampling estimator is used. Douc et al. [44, 45] showed that given a family of fixed proposal distributions, as in Green's generalization of Metropolis-Hastings, the mixture proposal can be suitably adapted at each iteration by adjusting the mixture weights. The prescribed adaptation scheme continually updates the mixture weights to reflect the relative

importance of the respective proposal distributions. It was shown to not only achieve the desired variance reduction but also lead to an asymptotic reduction in the Kullback-Leibler distance between the target and proposal distributions.

Their algorithm is known as Rao-Blackwellized D-kernel PMC. Given a family of proposal distributions, $q_d(\omega' \mid \omega)$, where $d \in [1..D]$, an initial proposal distribution, $q_0(\omega)$, and an unnormalized target distribution, $\pi(\omega)$, it produces a sample of size N and refines it over T iterations as follows:

1. Initialization ($t = 0$)

   (a) Generate the initial sample, $\left(\tilde{\omega}^{i,0}\right)_{1 \le i \le N}$, by drawing each point from the initial proposal distribution, q_0.

   (b) Compute the normalized importance weights:

   $$\mathrm{w}^{i,0} = \frac{\pi(\tilde{\omega}^{i,0})}{q_0(\tilde{\omega}^{i,0})} \Big/ \sum_{j=1}^{N} \frac{\pi(\tilde{\omega}^{j,0})}{q_0(\tilde{\omega}^{j,0})},$$

   for $1 \le i \le N$.

   (c) Resample $\left(\tilde{\omega}^{i,0}\right)_{1 \le i \le N}$ according to the importance weights, $\left(\mathrm{w}^{i,0}\right)_{1 \le i \le N}$, to get the corrected sample $\left(\omega^{i,0}\right)_{1 \le i \le N}$.

   (d) Initialize the mixture weights: c^0, where $c^0 \in \mathbb{R}^D$ and $c_d^0 = 1/D$ for $1 \le d \le D$.

2. For $t = 1..T$

   (a) Select a random proposal distribution for each sample point, $\left(q_{\tilde{d}}^{i,t}\right)_{1 \le i \le N}$, by drawing each $\tilde{d}$ from a categorical distribution with probabilities given by the mixture weight vector, c^t.

   (b) Generate a new sample, $\left(\tilde{\omega}^{i,t}\right)_{1 \le i \le N}$, by drawing each point from its respective proposal distribution, $q_{\tilde{d}}^{i,t}(\tilde{\omega}^{i,t} \mid \omega^{i,t-1})$.

   (c) Compute the normalized Rao-Blackwellized importance weights:

   $$\mathrm{w}^{i,t} = \frac{\pi(\omega^{i,t})}{\sum_{d=1}^{D} c_d^t q_d^{i,t}(\tilde{\omega}^{i,t} | \omega^{i,t-1})} \Big/ \sum_{j=1}^{N} \frac{\pi(\omega^{j,t})}{\sum_{d=1}^{D} c_d^t q_d^{j,t}(\tilde{\omega}^{j,t} | \omega^{j,t-1})},$$

   for $1 \le i \le N$.

(d) Resample $(\tilde{\omega}^{i,t})_{1\leq i\leq N}$ according to the importance weights, $(\mathrm{w}^{i,t})_{1\leq i\leq N}$, to get the corrected sample $(\omega^{i,t})_{1\leq i\leq N}$.

(e) Update mixture weights:

$$c_d^{t+1} = \sum_{i=1}^{N} \mathrm{w}^{i,t} p\Big(d \mid \tilde{\omega}^{i,t},\, \omega^{i,t-1},\, c^t\Big),$$

for $1 \leq d \leq D$, where:

$$p\Big(d \mid \tilde{\omega}^{i,t},\, \omega^{i,t-1},\, c^t\Big) = c_d^t q_d^{i,t}\Big(\tilde{\omega}^{i,t} \mid \omega^{i,t-1}\Big) \Big/ \sum_{l=1}^{D} c_l^t q_l^{i,t}(\tilde{\omega}^{i,t} \mid \omega^{i,t-1}).$$

Note that step 2(e) is based on the robust weight update proposed by Cappé et al. [23]. Iacobucci et al. [88] proposed a more robust important weight calculation that incorporates a second Rao-Blackwellizataion step to reduce the influence of the previous sample value proposal. Using double Rao-Blackwellization, step 2(d) can be replaced by:

$$\Big(\mathrm{w}^{i,t}\Big)_{1\leq i\leq N} = \frac{\pi(\omega^{i,t})}{\sum_{j=1}^{N} \mathrm{w}^{j,t-1} \sum_{d=1}^{D} c_d^t q_d^{j,t}(\tilde{\omega}^{i,t}|\omega^{j,t-1})} \Big/ \sum_{k=1}^{N} \frac{\pi(\omega^{k,t})}{\sum_{j=1}^{N} \mathrm{w}^{j,t-1} \sum_{d=1}^{D} c_d^t q_d^{j,t}(\tilde{\omega}^{k,t}|\omega^{j,t-1})}.$$

Given its advantages, PMC is the preferred approach to generating samples in associative tracking applications. This, above all else, is because it frees us from the concern that our sample size will be too small to yield sufficiently accurate estimates. In practice, we expect to be operating with a limited computational budget that restricts us to a predetermined maximum sample size. Even with well chosen proposal distributions, we prefer to be on firmer theoretical footing with regard to the convergence of our estimates. PMC also offers a distinct computational benefit, namely that the proposals for each sample at a given iteration can be drawn in parallel. However, for the purposes of this research we will still consider MCMC samplers as their performance will reveal a lot about our proposal distribution.

2.4.3 Proposal Distribution

For both the MCMC and PMC algorithms, the choice of the proposal distribution is a critical determinant of their effectiveness. A well-chosen proposal distribution

will efficiently explore the space, visiting all the high probability regions without lingering too long around any of its modes. When used for MCMC, such a proposal distribution will lead to rapid mixing of the chain and fast convergence of the ergodic averages. When used for PMC, a well-chosen proposal distribution will minimize the variance of the resulting estimates.

Unfortunately, there is no universal prescription for how to choose a good proposal distribution. It is a highly application-specific process and usually requires a fair amount of trial and error, which one of the criticisms of these algorithms. With that said, we adopt two principle strategies to assist us in constructing an effective proposal distribution for use with either MCMC or PMC. First, we use a proposal distribution that is capable generating large and meaningful changes in the state. Large is meant in sense that there is the potential to change the values of a significant fraction of the state variables. Meaningful, on the other hand, is meant in the sense that the candidate state is derived by altering the current state in a way that respects the structure of problem. Finally, we make the proposal distribution conditional on some function of the data in order to steer the chain toward regions that are likely to have a high posterior probability. This is known as a data-driven proposal. The idea was first formally introduced as a way speed up convergence in another high dimensional problem: image segmentation [186]. The proposal distribution $q(\omega' \mid \omega)$ is thus replaced by $q(\omega' \mid \omega,\ \tilde{Y})$, which is ideally defined to yield a distribution that is proportional to the posterior probability, over the support $q(\omega' \mid \omega,\ \tilde{Y})$.

Let $\omega = (Z,\ \mathit{Z0},\ \lambda,\ \alpha)$ be the state space for the joint posterior distribution, which consists of the indicator variables (cluster labels and false alarm indicators), the false alarm rate, and the concentration parameter, respectively. The state space is then partitioned into three parts and the proposal distribution $q(\tilde{\omega} \mid \omega^{(t-1)})$ is defined as the product of the componentwise proposal distributions:

$$q_\lambda\Big(\tilde{\lambda},\ Z^{(t-1)},\ \mathit{Z0}^{(t-1)},\ \alpha^{(t-1)} \mid \omega^{(t-1)}\Big) \tag{2.29}$$

$$q_\alpha\Big(\tilde{\alpha},\ Z^{(t-1)},\ \mathit{Z0}^{(t-1)},\ \lambda^{(t-1)} \mid \omega^{(t-1)}\Big) \tag{2.30}$$

$$q_z\Big(\tilde{Z},\ \tilde{\mathit{Z0}},\ \lambda^{(t-1)},\ \alpha^{(t-1)} \mid \omega^{(t-1)}\Big)\,. \tag{2.31}$$

Using this factorization, we have separated the relatively simple tasks of updating the false alarm rate and the concentration parameter from the considerably more difficult task of updating the indicator variables. Recall that Z and $\mathit{Z0}$ are each sets con-

taining nm variables. It is the typically large number of these indicator variables and the fact that there are strong dependencies between them that makes efficiently simulating the posterior distribution a challenging problem. At a minimum, the dependencies between the cluster labels and the false alarm indicators introduced in the definition of the prior require that each z_i and $z0_i$ pair be sampled jointly in order to keep their values in sync and avoid zero probability assignments. When operating on one variable at a time, the sampler cannot make state transitions that pass through zero probability regions of the space. However, rather than updating the indicator pairs one at a time, we instead jointly sample the entire block of indicator variables and update them all in a single step. This enables us to both ensure that the cluster labels and false alarm indicators remain in sync, and make proposals that differ substantially from the current state.

The Gibbs sampler paradigm suggests that (2.29)–(2.34) be defined using the full conditional distributions:

$$p\Big(\tilde{\lambda} \mid Z^{(t-1)},\, Z0^{(t-1)},\, \alpha^{(t-1)},\, \tilde{Y}\Big) \propto p\Big(Z0^{(t-1)} \mid \tilde{\lambda}\Big)\, p\Big(\tilde{\lambda} \mid a_\lambda,\, b_\lambda\Big) \tag{2.32}$$

$$p\Big(\tilde{\alpha} \mid Z^{(t-1)},\, Z0^{(t-1)},\, \lambda^{(t-1)},\, \tilde{Y}\Big) \propto p\Big(Z^{(t-1)} \mid Z0^{(t-1)},\, \tilde{\alpha}\Big)\, p\Big(\tilde{\alpha} \mid Z0^{(t-1)}\Big) \tag{2.33}$$

$$\begin{aligned} p\Big(\tilde{Z},\, \tilde{Z0} \mid \lambda^{(t-1)},\, \alpha^{(t-1)},\, \tilde{Y}\Big) \propto\; & p\Big(\tilde{Y} \mid \tilde{Z},\, \tilde{Z0}\Big)\, p\Big(\tilde{Z} \mid \tilde{Z0},\, \alpha^{(t-1)}\Big) \\ & p\Big(\alpha^{(t-1)} \mid \tilde{Z0}\Big)\, p\Big(\tilde{Z0} \mid \lambda^{(t-1)}\Big)\,, \end{aligned} \tag{2.34}$$

respectively. Of course, none of these distributions have a well-known form that allows for convenient direct sampling. However, they still provide a useful guide.

Since the false alarm rate, λ, and the concentration parameter, α, are scalar random variables, using univariate grid approximations to sample (2.32) and (2.33) provides one tractable approach to defining (2.29) and (2.30) based on the Gibbs sampler paradigm. When grid approximations are not desirable, a simple alternative is to define each of the proposal distributions as a Gaussian, left-truncated at zero, and centered on the current value. The truncation reflects the fact that $\lambda \geq 0$ and $\alpha > 0$. Note that the case of $\tilde{\alpha} = 0$ must handled correctly; the sample rejected or assigned a weight of zero, depending on the sampling algorithm.

An appropriate variance for each of the Gaussian proposal distributions can be derived from a rough gamma approximation to the full conditional whose parameters are a function of the conditional variables (and the known hyperparameters). The conditional probability of λ, given by (2.32), is approximately described by a gamma

distribution with shape and rate parameters $a = a_\lambda + nfa$ and $b = b_\lambda + ns$, respectively. The conditional probability of α, given by (2.33), is approximately described by a gamma distribution with shape and rate parameters $a = a_\alpha + a_k - 1$ and $b = b_\alpha + b_k$, respectively, where a_α and b_α are the parameters of the hyperprior for α, computed from $\mathcal{Z}0^{(t-1)}$, and a_k and b_k are the parameters of a gamma approximation to $p(Z^{(t-1)} \mid \mathcal{Z}0^{(t-1)}, \tilde{\alpha})$. That last approximation is based on rewriting the probability of the target cluster labels as $p(K \mid \mathcal{Z}0^{(t-1)}, \tilde{\alpha})$, which is the distribution of the number of clusters that it implies. As a function of α, with K fixed to the number of clusters in $Z^{(t-1)}$, this can be approximated as a gamma distribution whose mode is equal to the value of concentration parameter at which the expected number of clusters is K and whose variance is equal to the variance of the number of clusters when alpha is at the mode. The approximation can be computed using the modified mode and variance matching described in appendix A. Note that in practice we have found these gamma approximations to be good enough to be used effectively as independent (i.e. not conditional on the current state) proposal distributions. This provides yet another approach to defining (2.29) and (2.30).

To effectively sample the block of indicator variables, $(\tilde{Z}, \tilde{\mathcal{Z}}0)$, grid-based approximations would generally be intractable due to the dimensionality and approximations based on well-known distributions are not an option. It is here that we turn to the idea of a data-driven proposal distribution. However, before getting into the data-driven aspects, we first outline the basic form.

The proposal distribution (2.31) is itself defined as a mixture of proposal distributions carefully selected to generate candidate states by moving away from the current state in a variety of different ways so as to enable the efficient exploration of the entire sample space. Each mixture component is defined over some subset of the sample space that is both a function of the specific proposal and the current state. The family of proposal distributions used in the mixture consists of eight different moves:

$$\mathcal{M} = (1{:}\text{Add},\ 2{:}\text{Delete},\ 3{:}\text{Split},\ 4{:}\text{Merge},\ 5{:}\text{Extend},\ 6{:}\text{Reduce},\ 7{:}\text{Switch},\ 8{:}\text{Replace})\,.$$

Thus, (2.31) is written as:

$$\sum_i c_i q_i\Big(\tilde{\omega} \mid \omega^{(t-1)},\ \tilde{Y}\Big)\,, \tag{2.35}$$

where the mixture weights, c_i, and the mixture proposal distributions, $q_i(\cdot)$, are defined over the moves in $\mathcal{M}$.

In the context of MCMC, since each move is not necessarily reversed using the same type of move, we introduce the corresponding reverse moves:

$$\mathcal{M}^{-1} = (1\text{:Delete},\ 2\text{:Add}, 3\text{:Merge}, 4\text{:Split},\ 5\text{:Reduce},\ 6\text{:Extend},\ 7\text{:Switch},\ 8\text{:Replace}),$$

which are a permutation of $\mathcal{M}$. Our earlier notation for Green's modification to the Metropolis-Hastings acceptance probability must then be updated to make this explicit. The acceptance probability in this case is given by:

$$a\Big(\tilde{\omega},\, \omega^{(t-1)}\Big) = \min\left\{1,\ \frac{\pi\Big(\tilde{\omega}\Big)\, c_{\sigma(i)}\Big(\tilde{\omega}\Big)\, q_{\sigma(i)}\Big(\omega^{(t-1)} \mid \tilde{\omega},\ \tilde{Y}\Big)}{\pi\Big(\omega^{(t-1)}\Big)\, c_i\Big(\omega^{(t-1)}\Big)\, q_i\Big(\tilde{\omega} \mid \omega^{(t-1)},\ \tilde{Y}\Big)}\right\}, \tag{2.36}$$

where the weights are state-dependent and $\sigma(i)$ is the permutation mapping the index of the forward move to the corresponding reverse move.

In the context of PMC, the mixture proposal distributions and mixture weights apply to the D-kernel algorithm without any qualifications.

Each of the mixture proposal distributions, $q_i(\tilde{\omega} \mid \omega^{(t-1)},\ \tilde{Y})$, will be discussed in detail below. First we must address the broader issue of what exactly it means to be data-driven proposal. In effect, conditioning the proposal distributions on $\tilde{Y}$ does not really change anything. Using the Gibbs sampler paradigm as a guide already suggests that we use this information as we can see from (2.34). Whatever the rationale, the conclusion is that any proposed $(\tilde{Z},\ \tilde{Z0})$ should explain some subset of the data well in addition to being consistent with the prior. Although they stated it differently, Tu and Song-Chun Zhu [186] basically suggested that an effective way to make proposals that are likely to explain meaningful features in the data is to use the outputs of discriminative models to inform the proposal distribution. In their image segmentation application, they used the output of clustering and edge detection algorithms that typically produce, at least locally, decent segmentations to inform the proposed changes to the region and boundary parameters. We too will employ this approach to inform the mixture proposal distributions by approximating the likelihood term in (2.34) using the output of a discriminative model. So, what are the analogs of edge detection and clustering in the tracking domain? The tracklets and the feasible graph, which will be introduced next.

2.4.3.1 Feasible Graph

In image segmentation applications, the output of an edge detector provides local information about the likely boundaries between objects and regions. The output of a clustering algorithm provides local information about which pixels are likely to be part of the same object or region. When used to inform the proposal distributions, this information makes it possible to efficiently sample a generative model of the entire image. In tracking applications, such local information comes from identifying the pairwise associations between measurements that we call *tracklets*.

As the name suggests, a tracklet is simply a small track; consisting of just two measurements. It is the output of a discriminative model for the probability that the two measurements originated from the same target. In other words, the tracklet model describes:

$$p(A_{ij} \mid \tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j)\,, \tag{2.37}$$

where $A_{ij} \in \{0, 1\}$ is the event that the i-th and j-th measurement are associated. The association event can also be stated as: $A_{ij} = \{z_i > 0 \wedge z_i = z_j\}$. Tracklets can thus be seen as providing local information about which measurements are likely to belong to the same track.

The model (2.37) is referred to as discriminative, as opposed to generative, because it does not describe how the observed data is probabilistically generated from the unobserved association but rather directly describes the dependence of the unobserved association on the data. A generative tracklet model, on the other hand, would be based on the joint probability, $p(A_{ij}, \tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j)$, and then use Baye's rule to calculate $p(A_{ij} \mid \tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j)$. This, of course, would require us to explicitly model the likelihood, $p(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j \mid A_{ij})$, and prior, $p(A_{ij})$.

When the problem does not require the ability to draw samples from the joint distribution, then it can be preferable to use a discriminative model [132]. For our application, the preference for a discriminative model is largely motivated by the fact that it considerably simplifies the construction of rich tracklet models. Simple models based solely on speed constraints can be readily formulated as generative models. But, as we seek to build increasingly complex tracklet models, such as those that incorporate terrain constraints, it becomes costly and sometimes unintuitive to stay within the generative framework. The bottleneck is the likelihood, which becomes highly nonlin-

ear. Note that the tracklet model need not reflect the same constraints as the target likelihood model. It could, for example, consider the passability of the terrain between the measurements despite the fact that impassable barriers cannot be described using Gaussian likelihood models. At any rate, using a generative model to compute (2.37) presents no problem. We are, however, free to stop thinking (and modeling) in those terms whenever we see fit to do so.

The tracklet concept was originally developed as part of an effort to mitigate false alarms in a preprocessing stage used in conjunction with traditional tracking algorithms. In high false alarm environments, measurements with no plausible associations could be disregarded in order to avoid overloading the tracker and reduce the number of spurious tracks. For this purpose, tracklets performed extremely well. Although, it soon became clear that they had much more potential.

Given a set of measurements, $\tilde{Y}$, the set of all tracklets computed over those measurements with probability greater than τ can be used construct the *feasible graph*. The feasible graph is a directed acyclic graph (DAG) with vertices corresponding to the measurements and edges corresponding to the tracklets. In other words, an edge indicates that the measurements corresponding to the source and target vertices are likely to be part of the same track. The direction of each edge is determined by the time order of the associated measurements. Each edge is assigned a weight equal to the tracklet probability.

Figure 2.5 shows an example of a simple feasible graph based on 17 measurements collected over 7 scans. The graph is visualized in trellis form to make the scan constraint (discussed earlier) apparent. The weights are not shown. Feasible graphs can also be visualized by plotting each vertex at the position of the corresponding measurement. Figure 2.6 shows such an example based on a 16 minute data set containing two walkers in a moderate false alarm environment. The edges in part (a) are rendered without arrow heads to minimize the visual clutter.

Assuming that the tracklets have been computed correctly, each track consists of a (directed) path in the feasible graph. From this perspective, the tracking problem becomes one of assigning the vertices of the graph either the false alarm label or target cluster label. Although still a difficult problem because of the size of the solution space, it is considerably simpler than trying to search the entire space of associations. By making ($\tilde{Z}$, $\tilde{Z}0$) proposals that are informed by the edges and their weights, we can

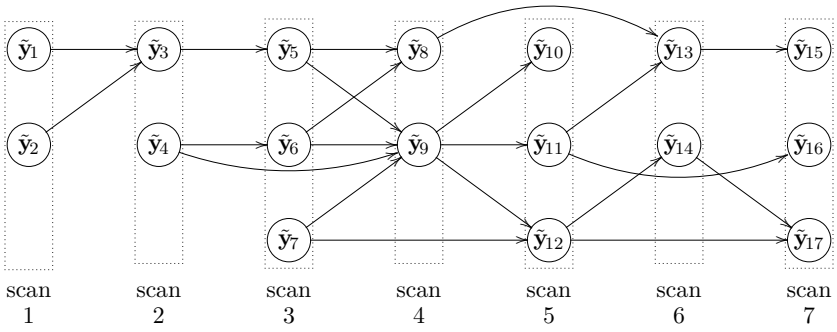


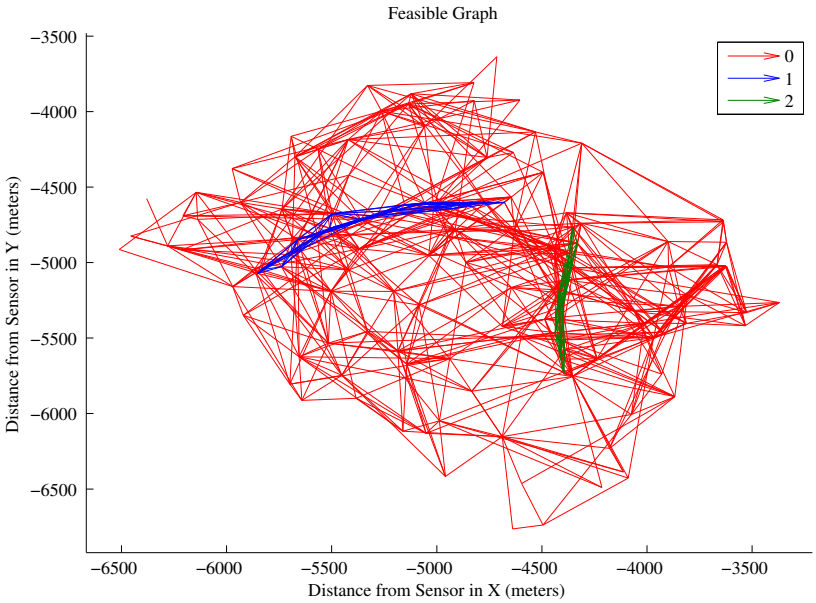
Figure 2.5: Example of a feasible graph in trellis form.

effectively bias the sampler's search toward regions of the state space that we expect to have a high probability under the model. Biasing the search in this manner is standard practice for complex stochastic search problems.

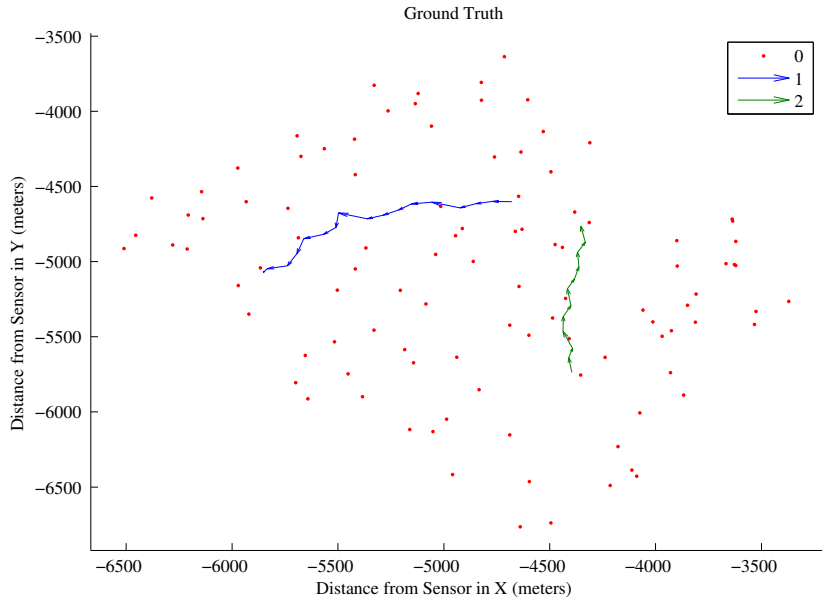
The feasible graph can thus be seen as one approach to mapping out the important structure in the space of associations. The tracklets identify the high probability subspaces using a tractable first-order association model and the feasible graph encodes their relationships. By combining the weights of related edges, we can approximate the likelihood of higher order associations.

The feasible graph can also be seen as a form of gating. Gating is a screening mechanism used to determine which measurements are valid candidates when updating the state of a given track. It has long been standard practice in the tracking community to disregard measurements whose distance to their predicted values is too large in some sense. This process is inherently generative as it requires the ability to estimate a measurement from the unobserved target state. The basic principle of ignoring associations deemed to be implausible based on some metric, however, remains the same. Of course, in traditional tracking algorithms these gating decisions are made sequentially with respect to the current estimated target state, which may quite possibly be inaccurate.

The idea of applying an association criteria directly to the measurements themselves, as in the feasible graph, is not new. Even the idea of combining these associations into a graph to be used by the tracking algorithm is not new either. Both of these concepts appeared in one of the first comprehensive Monte Carlo multiple target tracking algorithms from Oh et al [135]. Using their terminology, the graph is a neighborhood tree of observations. Its edges are unweighted and identified by applying a fixed thresh-



(a) Feasible graph in which the edges are colored based on their true cluster label.



(b) True associations.

Figure 2.6: Example of a feasible graph in its spatial form.

old to the Euclidean distance between pairs of measurements, without regard to their uncertainty. The number of edges in the graph is also limited by enforcing a maximum association distance which they compute based on the expected number of consecutive missing observations. Their multiple scan algorithm employs an MCMC sampler that does in fact use the neighborhood tree to inform proposals, but just those that reassign false alarm measurements to targets (e.g. add and extend moves). This is in fact a rudimentary example of data-driven sampling. Our proposed approach builds on these concepts by using more sophisticated association models to identify the edges of the graph and more aggressively exploits the information contained in the graph for the proposal distributions.

What exactly are the benefits of using more sophisticated association models? The obvious one is to get meaningful edge weights with which to approximate the measurement likelihood of any proposal. Another benefit is to handle hard terrain constraints in a way that does not rely on the likelihood model accurately reflecting the constraint. This is especially important in the usual case in which we employ linear-Gaussian likelihood models, or likelihood models based on linear-Gaussian approximations, because strictly satisfying arbitrary hard constraints is impossible using symmetric distributions with infinite spatial support; even constructing reasonable approximations is difficult at best. For example, consider two walkers moving in close proximity but on either side of an impassable barrier. Using a discriminative tracklet model, it is relatively easy to determine if the association of two measurements is precluded by such intractable terrain and simply return a zero probability so that the feasible graph will not contain the edge.

Now, in order for the feasible graph to be effective, it must contain edges between the actual consecutive target measurements. Whether or not this is the case depends in large part on the tracklet model. However, the difficulty in specifying a good tracklet model is no greater than what is already required in order to specify an accurate dynamical model for the likelihood. Both are predicated on understanding how targets move. If we lack such an understanding or the ability to formulate it probabilistically, then never mind the feasible graph; the whole tracking model itself is rather dubious. At any rate, a tracklet model can always be derived from the dynamical model. Of course, the fact that tracklets require only discriminative models affords us considerable freedom to incorporate knowledge about a target's behavior that would be too difficult

to represent using a likelihood function subject to distributional constraints.

Whether the feasible graph contains the necessary edges also depends on the tracklet probability threshold, τ, used to create it. We must always exercise caution when applying a fixed threshold because it entails throwing away information. To avoid rejecting any critical edges in the process, the threshold should be set as low as possible, where the desire to retain the requisite structure is being traded off against the computational cost of processing an extremely dense graph. Choosing a reasonable value is an application-specific question. However, we can often use simply $\tau = 0$. That is because we have other ways to control the growth of the graph that are less problematic.

The first is by enforcing a maximum association time. This defines the support of the model with respect to time. Exactly how far apart two consecutive measurements may be and still lie within the support of the model will depend on how constrained a target's motion is, which is both application-specific and possibly a function of the measured values. Targets with limited maneuverability can be given a larger maximum association time because there is less uncertainty regarding what they can do between observations. When the maneuverability is due to terrain constraints, for example, this value will depend on the measured position of the target. The second way that we can control the size of the feasible graph, independent of the tracklet probability threshold, is to limit the out-degree of any vertex. This can be implemented by only keeping the n-best outgoing edges. In practice, we have found that 10 works well for the radar tracking problems that we have studied, even when the false alarm rate is high.

For practical applications, the amount of time required to compute the tracklets and construct the feasible graph might be a concern. We have just mentioned a few ways to control the size of the graph, but only the maximum association time will help speed up the construction of the graph. Although the threshold value and limits on the out-degree will produce graphs that are smaller and hence less costly to process, the tracklet probability must still be evaluated before any potential edges can be rejected so we incur that cost regardless. Fortunately, when time is of the essence we can exploit the fact that the tracklet processing lends itself to parallelism because each measurement pair can be evaluated independently. It is also worth keeping in mind that the whole motivation for the feasible graph was to reduce the sampling time required to achieve a given accuracy by enabling us to maximize its efficiency through better proposals.

2.4.3.2 Mixture Proposals

Before describing each of the distributions in (2.35), let us first consider the broader issue of how we can use the feasible graph to inform the proposals. We begin by recalling that we would like each proposal to be proportional to (2.34), which we restate here for convenience:

$$p\Big(\check{Y} \mid \tilde{Z},\, \tilde{Z0}\Big)\, p\Big(\tilde{Z} \mid \tilde{Z0},\, \alpha^{(t-1)}\Big)\, p\Big(\alpha^{(t-1)} \mid \tilde{Z0}\Big)\, p\Big(\tilde{Z0} \mid \lambda^{(t-1)}\Big)\,. \tag{2.38}$$

From this we can see that there are four factors that should influence the distribution of the proposed indicator values: the likelihood of the measurements, the prior distribution over the target cluster labels, the concentration parameter hyperprior, and the prior distribution over the false alarm cluster labels. It is the dependence on the measurements in the first factor, the likelihood, that leads to the notion of being data-driven.

For each move, we would ideally like to enumerate all possible outcomes and weight them using the product of all four factors. Although this is a realistic option for the delete move, it would generally be intractable for the add, extend, and replace moves because of the combinatorial explosion in the number of possible ways those moves could be made. In the case of the remaining moves, it is conceivable that we could do this but the cost of the additional likelihood evaluations (beyond those required by the sampler) would likely become prohibitive. For multiple model systems and those with complex nonlinear dynamical models, which we expect to be commonly used in practice, computing the likelihood is a relatively expensive operation. Fortunately, the feasible graph provides a means of addressing both of these difficulties enabling us to efficiently generate proposals from an approximation to the desired distribution.

The contribution of the likelihood factor in (2.38) to the probability of any proposed $(\tilde{Z},\, \tilde{Z0})$ is to represent the degree of agreement between the measurements and the associations implied by those indicators. That agreement can also be quantified by using the edge weights in the feasible graph as a proxy for the likelihood of the measurements. This is because, given accurate likelihood and tracklet models, the marginal likelihood of two consecutive target measurements, conditional on their assignment to the same track, is proportional to the association probability.

By treating the likelihood of each pair as independent, we can use the product of the corresponding edge weights to approximate the likelihood for a single target. Based on our earlier assumptions, the likelihood of each target is independent so we

can combine these estimates via their product. The likelihood of the false alarm measurements, on the other hand, can be approximated in a couple of different ways. One approach would be to take the product of the complementary probabilities of the edges that do not link two vertices assigned to the same target (i.e. the edges not used in calculation of the target likelihoods). Another approach would be to use one per-vertex cost to approximate the likelihood of each false alarm measurement, which would match the uniform false alarm model. These lightweight likelihood approximations can be used by any of the moves.

For the add, extend, and replace moves, we can exploit the directed acyclic structure of the feasible graph, in conjunction with the likelihood approximation, to estimate the weights of the proposed indicator values without enumerating all of the possibilities. In all cases, the proposal probabilities will reflect the contribution of the first and last terms in (2.38). Whether or not we can factor in the effect of the second and third terms will depend on their specific forms, which we will elaborate on later.

A precise definition of all the moves will be provided below. For now, suffice it to say that these three are the most complex because they each entail reassigning false alarm measurements to a target label. This is equivalent to finding a path in the subgraph of false alarm measurements, which we refer to as the feasible false alarm graph. Since enumerating and weighting the possible paths would be intractable for any realistic graph, we instead settle for an approximate solution based on estimating the average weight for all possible paths of each possible length, which is tractable even for large graphs, and a mechanism for constructing paths of a given length.

To do this, we need a way to estimate the average weight for all possible paths of each possible length, from a given starting vertex. Of course, it must be possible to reliably construct these paths for a given length.

We begin by constructing a cost function that reflects the four factors in (2.38) and that can be computed incrementally over a path. This cost represents the estimated weight of the path.

The contribution of the likelihood is computed using the lightweight approximation derived from the edge weights. Its product form makes it amenable to the incremental processing of the cost function since we can simply subtract the log cost of the current configuration and add in the log cost of the change at each step during processing.

The contribution of the prior distribution over the false alarm cluster labels can easily be included without approximation since it naturally has a product form and requires relatively lightweight computation. We need only compute the cost of the current number of false alarms in each scan, and the cost of each scan with one fewer false alarm (for scan containing at least one false alarm). Thus, the computational expense is a function of the number of scans and is not impacted by the density of the graph. As before, we can subtract the log cost of the current configuration, for the scan in which the measurement associated with the given vertex was made, and add in the log cost of the configuration with one fewer false alarm. Since the feasible graph structure reflects the scan constraint, any vertices in the feasible false alarm graph that lie along a given path must be in different scans.

Only certain forms of the prior distribution over the target cluster labels can be directly incorporated into a proposal cost computed incrementally over a path. Fortunately, those make up an important class of priors: exchangeable partition priors. Their contribution can be included without approximation as well. Such priors will be discussed in detail in the next chapter, so for now we will state why that is the case without further explanation. Exchangeable partition priors do not require the individual assignments in order to be evaluated. They require only the vector of cluster sizes. This allows us to evaluate them without knowing which vertices are in each cluster. Their contribution to the cost is added in whenever the cost for a given length is finalized by adding the given length to the cluster size vector. Thus, the computational expense of this contribution is a function of the number of path lengths being considered, whose maximum value is just the number of scans.

Other forms of the prior distribution must be examined on a case by case basis. Depending on the prior, it may be possible to represent some of preferences that it expresses for different associations in the form of edge or vertex weights that can be computed incrementally over a path.

Finally, the contribution of the concentration parameter hyperprior can be can be included without approximation when the prior distribution over the target cluster labels is exchangeable and we compute $p(\alpha^{(t-1)} \mid \tilde{Z0})$ without regard to the scan constraint. In the next chapter, more will be said about what exactly this means. For suffice it to say that exchangeable priors assign nonzero probability to associations that are invalid because they violate the scan constraint. When evaluating the prior, this

does not pose a problem. However, it does impact the concentration parameter. We can correct, to some extent, for the impact on α of re-normalizing the prior probability over the valid associations but this requires some knowledge of the maximum scan depth. Again, more will be said about this in the next chapter. In terms of these cost calculations, we can approximate the contribution of the hyperior when scan constraint is observed by noting that for any path, after reassigning a given set of false alarm measurements, the maximum scan depth either remains the same or increases by one. In parallel with our incremental cost calculations, we can keep track of the expected maximum scan depth at each step. This contribution to the cost is also added whenever the cost for a given length is finalized. Thus it has the same computational expense as the prior distribution over the target cluster labels.

Using this cost function, the average cost of all possible paths starting at a given vertex can be estimated as follows.

At the initial vertex, initialize the bookkeeping which is associated with every vertex visited during the course of the processing. This consists of a set containing the lengths of all possible paths to the initial vertex along with a weight and count corresponding to each length. The weights indicate the total weight of all paths of their respective lengths. The counts indicate the number of paths of their respective lengths. Here, the only length in the set is zero and its count is one by definition. The weight is initialized with the cost contribution from the prior distribution over the false alarm cluster labels based on adding the initial vertex to the path and thus assigning it a target cluster label. Note that there is no initial likelihood contribution. This is because a single target vertex is assumed to make no contribution to the target likelihood; equivalent to an implicit probability of one term in the product.

The processing proceeds by visiting all the vertices that are reachable from the initial vertex in topological order. Note that the graph has a natural topological ordering since the vertices correspond one-to-one with measurements that are time ordered. A given vertex is reachable if and only if the source vertex of at least one incoming edge has already been visited. A vertex has been visited when the set of lengths associated with it contains at least one element.

At each vertex j, we update its bookkeeping incrementally for each incoming edge whose source vertex, i, has already been visited. Each such edge, $i \to j$, connects the current vertex to the initial vertex through at least one path. The number of paths

is given by the sum of counts recorded at vertex i. The lengths of those path are given by the set of lengths associated with vertex i, plus one. The lengths, weights, and counts associated with vertex i will be used to update the corresponding values for vertex j.

For each edge, $i \to j$, compute the incremental cost of adding j to the path based on the contributions from the likelihood and the prior distribution over the false alarm cluster labels. Let that incremental cost be w_{ij}. Now, iterate over the path lengths associated with vertex i. A path of length L to vertex i corresponds to a path of length of $L+1$ to vertex j. If this is the first time we are encountering length $L+1$, then we add that length to the set associated with vertex j. Its weight is initialized to corresponding weight from vertex i plus w_{ij}. Its count is initialized to the corresponding count from from vertex i. If we have already seen length $L+1$, then we increment the weight and count corresponding to that length in the set associated with vertex j. The weight is incremented by the corresponding weight from vertex i plus w_{ij}. The count is incremented by the count from from vertex i.

Once all the incoming edges have been processed and the bookkeeping at vertex j is up to date, every path length in the set can be finalized. This is accomplished, for each length, by first computing the average cost using the weight and count corresponding to the given length. Then the cost contributions from the prior distribution over the target cluster labels and concentration parameter hyperprior are added. The final result is an estimate of the average weight of all possible paths from the initial vertex to vertex j, for the given length. These estimates can then be aggregated over all the reachable vertices to get an estimate of the weight for all possible paths from the initial vertex.

The bookkeeping associated with each vertex can then be used to reconstruct a path of the desired length, L, that begins at the initial vertex. This process starts by identifying the vertices at which a path length of L was recorded. Using any suitable means, one of those vertices is selected to be the final vertex in the path. The process then continues by examining the incoming edges to find a source vertex at which a path length of $L-1$ was recorded. There may be multiple choices and one can be selected using any suitable means. The selected vertex is added to the path. The process repeats in this manner until the final $L=1$ selection has been made.

At long last we have the two tools that are needed to efficiently generate the proposals for each move.

Add The add move modifies the current state by introducing a new target. It is the first of two moves designed to increase the number of targets. The move is made by selecting two or more vertices from among those assigned to the false alarm cluster and then reassigning them to a new target label. The selected vertices must constitute a path in the feasible graph.

This is the most complex because the number of possible ways to do this is generally a very large number. The proposal distribution is thus based on an approximation which reflects, to the extent possible, the influence of all four factors in (2.38). Here, we employ the previously described procedure for estimating the average weight for all possible paths of each length, from a given vertex, and the procedure for constructing the paths. The estimation procedure, however, assumes that we have the starting vertex for the path, which is not the case for an add move.

We could, of course, select one of the false alarm vertices to initiate the path. This could be done uniformly at random or by using the false alarm model to favor making the selection from a scan with more than the expected number of false alarms. Given the starting vertex, we can choose the length of the path based on the average weight and then follow the prescribed procedure to construct the path. Note that the path length weight calculations and the construction procedure can performed either forward or backward in time (by implicitly reversing the edges in the graph) so that selections made near the end of a true path will not always result in short fragments.

Although this approach is indeed viable and relatively efficient, it tends to produce shorter paths compared to approaches that select the starting vertex itself with consideration for the possible paths. Since our goal is to engineer an add move that makes large and meaningful proposals, we prefer approaches of the latter type despite their increased computational cost. To implement such a move, there are two ways that we can proceed.

First, we can simply run the estimation procedure repeatedly using each false alarm vertex as the starting point. In practice, this is possible since the cost is $O(n \log n)$ in the number of false alarm vertices. Using this approach, we aggregate the weights and counts for each possible path length over all potential starting points. These are then turned into averages and normalized to become the distribution over path lengths. Once a path length is chosen, we select the starting vertex from among the vertices that contain a path of the given length. This can be easily determined when we record the

longest path from each initial vertex during the estimation procedure. The distribution over the possible starting vertices is based on the average weight estimates. Next, we select the end vertex from among those reachable from the starting vertex at the given length. The distribution over these selections is again based on average weight estimates. From here, we use the path construction procedure to select the remaining vertices. At each step in this procedure, the distribution over the possible selections is based on the average weight estimates.

The second way is to run the estimation procedure for each false alarm vertex that has no incoming edges whose source vertices are false alarms. In other words, the estimation procedure is run for each source vertex in the false alarm feasible graph. At each of these potential starting points, we record the longest path. Thus, for a given potential starting point we know the support of the possible path lengths that emanate from it and can compute the average weight for each one. However, what we do not have are the average weights for any intermediate path, which are legitimate candidates for proposals as well. We can approximate those weights and counts for each possible path length using a sliding-window type approach. The basic idea is to compute the average weight of the paths of length L, from the given potential starting vertex, by taking the average weight of the paths of length $L + n$ and then subtracting off the average weight of the paths of length n. Care must be taken to add back in the average weight related to the vertex at length n that is lost in the subtraction but should be part of the weight estimate. This can be handled using some additional bookkeeping during the original estimation procedure. The resulting estimate is an average which we then scale up to a total weight using the count corresponding to the paths of length $L + n$. These sliding-window estimates are performed for each potential starting point, and the weights and counts for each possible path length are aggregated. These are then turned into averages and normalized to become the distribution over path lengths. In practice we have found that this approximation produces distributions over path lengths with the same basic shape as the first approach but with thinner tails because of the extra averaging. Figure 2.7 shows an example of how the distributions over path lengths computed using the two approaches differ. The example was generated from a feasible graph with 709 vertices and 2096 edges computed from 24 minutes of radar data. This loss of density at the tails might be undesirable for some applications, in which case the first approach must be used.

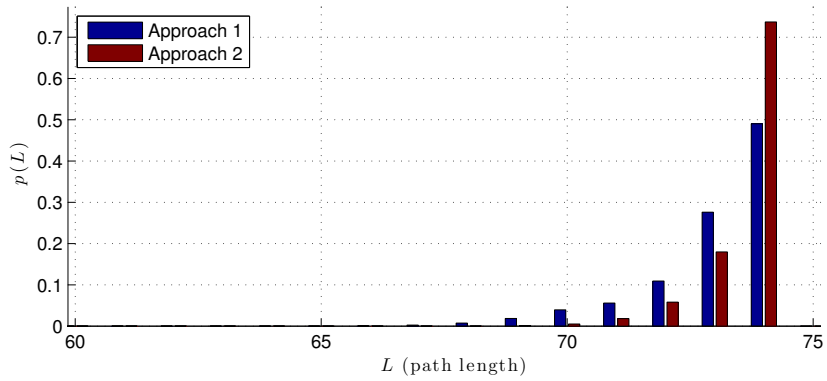


Figure 2.7: Example of the distributions over path lengths for an add move using the two different approaches to computing the weights.

Once a path length is chosen, this approach continues by selecting the starting vertex from among the vertices that contain a path of the given length as before. The distribution over the possible starting vertices is based on the average weight estimates, accumulated for each one from the sliding-window estimates. Next, we select the end vertex from among those reachable from the starting vertex at the given length. However, unlike in the previous approach, the possible candidates are all those vertices at the given length or more because we want to include the intermediate paths. Through some additional bookkeeping during the sliding-window estimates we can get the necessary averages weights to use for this distribution. From here, we use the path construction procedure to select the remaining vertices as before.

Note that when evaluating the probability of an add proposal generated using the second approach, we must be sure to properly account for the fact that there may be multiple start vertices that could have been used to construct the given path.

Note also that if desired, the path length can be selected without regard to the estimated weights. For example, we might select the length with a probability that is proportional to the length to reflect certain prior beliefs or perhaps uniformly at random to increase the variation in the moves. The path construction process remains the same in any event.

Delete The delete move modifies the current state by removing one of the targets. It is the first of two moves designed to reduce the number of targets. The move is made by selecting one of the target clusters and then reassigning its vertices to the false alarm

cluster.

This is perhaps the least complex move because the number of possible selections is equal to the number of targets. The proposal distribution is computed by enumerating the selections, one for each target, and then computing the weights based on all four factors in (2.38). Note that the likelihood approximation can certainly be used here, but in this case the full target likelihoods are already available from the last evaluation of the probability of the current state. Thus, to use the actual model for the likelihood contribution, we need only recompute the false alarm likelihood and then combine it with the likelihoods for the remaining targets.

A alternative formulation would be to simply select the cluster to be deleted uniformly at random.

Split The split move modifies the current state by splitting one of the target clusters into two. It is the second of two moves designed to increase the number of targets. In contrast to the first, the add move, it increases the number of targets without changing the false alarm assignments. The move is made by selecting one of the target clusters and one of its vertices to be the split point. The vertices that follow the split point in time are then reassigned to a new target label. Following the split, each target cluster must consist of at least two vertices.

The proposal distribution can be computed in one of two ways, depending on how expensive it is to evaluate the prior distribution over the target cluster labels.

The first is appropriate when an application seeks to minimize these prior evaluations. In this case, the cluster to be split is selected uniformly at random. The possible split points are then enumerated and their weights computed based on the contributions from the target likelihood, using the approximation, and the prior distribution over the target cluster labels.

For the second approach, the proposal distribution is based on enumerating all possible split points. The weights are computed in the same manner as the first approach.

Merge The merge move modifies the current state by combining two existing target clusters into a single cluster. It is the second of two moves designed to reduce the number of targets. In contrast to the first, the delete move, it does this without changing the

false alarm assignments. The move is made by selecting the two target clusters and then reassigning the vertices in one to match the label used by the other.

To make such a move requires an edge in the feasible graph linking the last vertex in one cluster to the first vertex in the other cluster. The proposal distribution is computed by enumerating the possible merges. The weights are computed based on the contributions from the target likelihood, using the approximation, and the prior distribution over the target cluster labels.

This move also presents an opportunity to potentially address the problem of a missing edge in the feasible graph. It would be possible to allow edge-free merges, assuming that the first vertex after the merge point follows the last vertex before it in time. Such proposed merges could potentially reduce the fragmentation of tracks caused by missing edges. The weights for the proposal distribution could either ignore the likelihood contribution or we could compute the association probability on the fly.

Extend The extend move modifies the current state by increasing the size of an existing target cluster (i.e. increasing the length of the track). It is the first of four moves designed to modify the current state in a way that does not change the number of targets. The move is made by first selecting one of the target clusters for extension. Then, one or more vertices are selected from among those assigned to the false alarm cluster and reassigned to the target label from the selected cluster. The selected vertices must constitute a path in the feasible graph that is contiguous with the path formed by initial target vertices.

Like the add move, the number of possible ways to do this is generally very large so we must employ the previously described procedure for estimating the average weight for all possible paths of each length, from a given vertex, and the procedure for constructing the paths. In this case, however, we do know the starting vertices: the first and last vertices in each track. We include the first vertices here so that extensions can take place forward in time as well. The estimation procedure works in reverse-time essentially as before but with out-edges considered instead of in-edges and the reverse topological order used to process the vertices.

A proposal is generated by running the estimation procedure using each potential starting vertex. We aggregate the weights and counts for each possible path length over all potential starting points. These are then turned into averages and normalized

to become the distribution over extension lengths. This is essentially how the first approach to the add proposal begins and we can just follow those steps to complete the extension.

Reduce The reduce move modifies the current state by reducing the size of an existing target cluster (i.e. reducing length of the track). It is the second of four moves designed to modify the current state in a way that does not change the number of targets. The move is made by first selecting one the target clusters for reduction. Then, one or more vertices from either end of the track are reassigned to the false alarm cluster label.

As was the case with the split move, the proposal distribution can be computed in one of two ways, depending on how expensive it is to evaluate the prior distribution over the target cluster labels.

For the first approach, the cluster to be reduced is selected uniformly at random. The possible combinations of reduction length and direction are then enumerated and their weights computed based on all four factors in (2.38).

For the second approach, the proposal distribution is based on enumerating all possible combinations of reduction length and direction. The weights are computed in the same manner as the first approach.

Switch The switch move modifies the current state by reassigning the labels from two targets so that at a certain point in each track, the first target receives the vertices previously assigned to the second target, and the second target receives those previously assigned to the first. It is the third of four moves designed to modify the current state in a way that does not change the number of targets. It is the only such move guaranteed to leave the number of false alarms unchanged.

To make such a move requires a certain local structure in the feasible graph which we refer to as a switch point. An example of this can be seen in figure (2.5) with vertices (measurements) 5, 6, 8, and 9. When vertices 5 and 6 are assigned to two different targets, and vertices 8 and 9 are assigned to the same two targets, then a switch is possible and the switch point is said to be valid. The move is made by reassigning the labels in each target cluster that follow the switch point in time. In this example, the reassignment would begin with the vertices 8 and 9 for their respective tracks.

The proposal distribution is computed by enumerating the possible selections,

one for each valid switch point. These can be easily identified when we flag potential switch points (i.e. the correct structure irrespective of the labels) during the construction of the feasible graph. In practice, we have found the number of possible selections to be quite modest. The weights are computed based on the contributions from the target likelihood, using the approximation, and the prior distribution over the target cluster labels. The other factors do not apply here since the false alarm assignments are not altered.

Replace The replace move modifies the current state by replacing one of the target clusters. In effect, it is like a delete move necessarily followed by an add move. It is the fourth of four moves designed to modify the current state in a way that does not change the number of targets. The move is made by selecting a target cluster for replacement uniformly at random. Its vertices are reassigned to the false alarm cluster and then the same steps for an add move are performed.

The proposal distribution for this move is not explicitly computed. In practice, both drawing and evaluating the probability of a replace move is based on the combination of the delete, with a uniform selection probability, and add moves.

2.4.3.3 Alternative Mixture Proposals for PMC

When using population Monte Carlo (PMC), we can consider an alternative definition of the components in (2.35) for applications in which we would be comfortable specifying a fixed upper bound, K, on the number of targets. In this case, the mixture would consist of $K+1$ components each defined using a self-reversing k-add move, where $k \in [0..K]$.

k-Add The k-add move modifies the current state by replacing any existing target clusters with k new targets. If $k = 0$, then the proposed state contains no targets. The move is effectively a deterministic deletion step followed by k add moves.

Thus, each distribution generates proposals with a fixed number of targets. The weight adaptation scheme in the PMC algorithm would then be responsible for determining the correct mixture weights. Those weights will eventually come to reflect the expected distribution of the number of targets.

2.4.3.4 Initialization and Boundary Conditions

To draw the initial state, we start by assuming that ($Z^{(t-1)}$, $\mathcal{Z}0^{(t-1)}$) indicates that all measurements assigned are the false alarm cluster. The initial values for λ and α can then be drawn from their respective proposal distributions, (2.29) and (2.30). Conditional on those values, the initial values for Z and $\mathcal{Z}0$ can be drawn using a slight modification to the add move. The move is modified to consider a path length of zero in addition to all possible nonzero path lengths. We can then repeatedly draw from the add move proposal distribution, each time using the most recent set of indicator variables as the current state, until a path of length zero is drawn. At that point, the initialization is complete.

So far, we have not said anything about what happens when a proposal fails. One way that this can happen is if the support of its proposal distribution is the empty set. For example, suppose we try to make a proposal using the delete move when there are no targets. This concern only applies to $q_z(\cdot)$ because not moves can be made from a given state. Another sort of failure mode is when a proposed value is invalid. This can happen when $q_\alpha(\cdot)$ returns precisely zero. Note that $q_\lambda(\cdot)$ cannot suffer either fate.

These boundary conditions need to be handled properly in the sampler. For MCMC, any move that fails will not be reversible and it should automatically be rejected. For PMC, this corresponds to assigning a weight of zero. Fortunately, for the all the moves in $q_z(\cdot)$ described above, it is possible check their validity (i.e. there exists at least one proposed state so they will not fail) without excessive effort before making any draws. We can simply zero out the mixture weights for any invalid moves and renormalize before proceeding.

Chapter 3

Cluster Label Priors

3.1 Introduction

In order to complete the model specification, we must define the prior distribution of the target cluster labels, $p(Z \mid \mathcal{Z}0, \alpha)$. This distribution implicitly represents our beliefs about the number of targets and is therefore crucial to the performance of the model. If $p(Z \mid \mathcal{Z}0, \alpha)$ does not adequately capture these beliefs, then we cannot expect the model to yield reasonable inferences. Unfortunately, this presents a major difficulty because the number of targets is typically unknown and, in fact, one of things that we wish to infer. In this case, the prior distribution should suitably represent our lack of knowledge and allow us remain agnostic about the specific number of targets.

We will assume that the number of targets is indeed unknown. Therefore, we require a prior distribution that is flexible enough to accommodate any number of targets which may arise. One compelling approach to specifying such a prior is through the use of Bayesian nonparametric models.

Instead of assuming that $p(Z \mid \mathcal{Z}0, \alpha)$ belongs to a given family of models indexed by a fixed dimensional set of parameters, we can define it using a nonparametric model on an infinite-dimensional parameter space. This rich class of models remains tractable despite having an unbounded number of parameters because only a finite subset of the available parameter dimensions are actually used to explain the data. The complexity of the model is adapted to fit the data during the inference process. Hence, nonparametric models constitute a data-driven approach to model selection which avoids fitting and then comparing multiple models of varying complexity. When the parameter space is chosen to be the set of distributions, the uncertainty about the prior distribution is described using a random probability distribution. Such models essentially express the prior as a stochastic process. In fact, Bayesian nonparametric prior models are typically named for the stochastic process that they represent.

Thus, a Bayesian nonparametric prior model allows us to avoid speculating about the number of targets and lets the data drive the decision. We will focus initially on a particular class of such models known as species sampling models because they are well-suited to the tracking application. Motivated by their structure, we will also consider a couple of different nonparametric prior models with different properties but that do not fully satisfy all the technical conditions necessary for a species sampling model. Finally, we will look at some alternative approaches to defining the prior distribution

of the target cluster labels for comparison.

3.2 Background

The question of how to handle the occurrence of novel phenomena, such as the appearance of a target, in Bayesian inference goes back as least as far as Bayes' theorem itself [202]. In the modern statistics literature, this question is well-studied in the context of *species sampling*.

From this perspective, we view Z as being generated through the random sampling of individuals from a population. Each element is an indicator variable, z_i, containing a label that identifies the species of the i-th sample. Targets then correspond to species that have been observed. Which species are present in the population is assumed to be a priori unknown.

The sequence $(z_1, z_2, \ldots)$ can be described by defining a *prediction rule* (or rule of succession):

$$p\left(z_{n+1} = c_j \mid z_1, \ldots, z_n\right),$$

where c_j is an arbitrary label assigned to the j-th distinct species to have been observed. Assuming that $k(n)$ distinct species have been encountered after sampling n individuals from the population, $j \in \{1, \ldots, k(n), k(n)+1\}$. The label $c_{k(n)+1}$ identifies a new species; one that we are observing for the first time.

The first thing to note about this formulation is that the support does not cover the set of all species. In other words, we cannot speak of the probability that sample $n+1$ is a particular species. This is a consequence of the fact that we do not know which species are present prior to observing them. At any point during the sampling process we might encounter an unknown and unsuspected species making it impossible to enumerate all of the possible outcomes. Therefore, z_{n+1} cannot be modeled using a distribution over species. The indicator variables must instead be defined based on what is known at the present sample time: all the species that have been observed in the previous samples, plus the additional possibility of seeing an unspecified new species. The support of $p(z_{n+1} \mid z_1, \ldots, z_n)$ is necessarily reduced to reflect these possible outcomes. Thus, the information conveyed by the distribution is not actually about individual species but rather about partitions of the variables where each part is the set $\{z_i \mid z_i = c_j\}$ for values of j corresponding to observed species. In other

words, the sequence $(z_1, \ldots, z_n)$ implicitly defines a random partition of the variables by clustering observations sharing a common label. Equivalently, we could view this as a partition of the index set $\{1, \ldots, n\}$ by only considering the indices. For example, suppose that after sampling five individuals we get the sequence:

$$(z_1 = c_1,\, z_2 = c_2,\, z_3 = c_2,\, z_4 = c_3,\, z_5 = c_1)\,.$$

The partition of the index set $\{1, \ldots, 5\}$ generated is:

$$\{1, 5\} \cup \{2, 3\} \cup \{4\}\,.$$

The next thing to note is that the labels themselves are arbitrarily defined and assigned based on the order in which new species are encountered[1], which should be regarded as unpredictable owing to the randomness of the sampling process. This implies that their particular values do not contain any useful information and the prediction rule should therefore be specified such that the resulting sequence is exchangeable with respect to the labels. In other words, the joint distribution, $p(Z)$, should be invariant under permutations of the labels.

Finally, it follows from the randomness of the sampling process that the order of the observations is arbitrary. This implies that the index values themselves are uninformative. The prediction rule should therefore be specified such that the resulting sequence is exchangeable with respect to the indices (i.e. "time") in addition to the aforementioned exchangeability with respect to the labels. Hence, the joint distribution should be invariant under permutations of both the labels and indices. The assumption of exchangeability with respect to the indices is the traditional notion of exchangeability popularized by de Finetti's representation theorem. It expresses our ignorance, in the sense that we cannot distinguish one z_i from another without the benefit of additional data, through a symmetry of beliefs that assigns the same probability to any ordering of the variables. The notion of exchangeability with respect to the labels is based on the idea that our ignorance about the possible outcomes (i.e. the labels) should result in a similar symmetry of beliefs. Probability functions that satisfy both notions of exchangeability are known as *partition exchangeable* [202].

[1] Although it is possible to retrospectively associate the labels with individual species after they have been observed, the values are in fact arbitrary since each new label must begin its existence representing the unspecified new species.

Consider a prediction rule of the form:

$$p(z_{n+1} = \hat{c} \mid z_1, z_2, \ldots, z_n) = \sum_{j=1}^{k(n)} p_j(\mathbf{m}(n))\, \delta_{c_j}(\hat{c}) + p_{k(n)+1}(\mathbf{m}(n))\, F_0(\hat{c})\,, \tag{3.1}$$

with $z_1 \sim F_0$. This is a mixture of point-masses at the labels corresponding to observed species and a nonatomic (diffuse) base distribution, F_0, that represents the distribution of new species labels. The weights of the mixture are given by the functions $p_j(\mathbf{m}(n))$, which satisfy the conditions:

$$p_j(\mathbf{m}(n)) \geq 0 \text{ and } \sum_{j=1}^{k(n)+1} p_j(\mathbf{m}(n)) = 1, \tag{3.2}$$

for all $\mathbf{m}(n)$. The argument $\mathbf{m}(n)$ is the cluster size (or frequency) vector given n observations, which contains $k(n)$ elements and is defined such that the j-th element is the number of times the j-th observed species appears in $(z_1, \ldots, z_n)$; that is, $m_j(n) = \sum_{i=1}^{n} I\,(z_i = c_j)$.

Note that since the labels are arbitrarily defined, it is convenient to assume that F_0 is a uniform distribution on the interval $[0, 1]$. Each observed species is thus assigned a real-valued label in $[0, 1]$ and the labels are almost surely distinct.

For $j \in \{1, \ldots, k(n)\}$, the functions $p_j(\mathbf{m}(n))$ give the probability that, after n observations with cluster sizes $\mathbf{m}(n)$, the next one will be the j-th observed species. The probability that the next observation will be a new species is given by the function $p_{k(n)+1}(\mathbf{m}(n))$. Collectively, these are known as the *prediction probability functions*. Note that they depend on the data only through the cluster sizes in $\mathbf{m}(n)$. When the prediction probability functions are defined such that the sequence described by (3.1) is exchangeable (with respect to the indices), the sequence is called a *species sampling sequence* [144]. In fact, an exchangeable sequence $(z_1, z_2, \ldots)$ is a species sampling sequence if and only if it is subject to a prediction rule of the form (3.1).

Not all functions $p_j(\mathbf{m}(n))$ that satisfy the conditions in (3.2) can be used to describe a species sampling sequence. In fact, most choices will not yield an exchangeable sequence. So, what can be said about the families of prediction probability functions that will guarantee the required exchangeability of $(z_1, z_2, \ldots)$? According to Johnson's 'sufficientness' postulate [203], one such family consists of functions of the form: $p_j(\mathbf{m}(n)) = f(\mathrm{m}_j(n), n)$. This assumes that the prevalence of one species conveys no information about the prevalence of others. See Lee et al. [111] for more details on the valid forms of prediction probability functions for species sampling sequences.

Given a species sampling sequence, it can be shown that as $n \to \infty$, the distribution $p(z_{n+1} = c_j \mid z_1, z_2, \ldots, z_n)$ converges, in total variation norm, almost surely to the random probability distribution [144]:

$$F = \sum_{j=1}^{\infty} w_j \delta_{c_j} + \left(1 - \sum_{j=1}^{\infty} w_j\right) F_0, \tag{3.3}$$

where the c_j are i.i.d. draws from F_0 (and independent of w_j), and the w_j are a sequence of weights, which satisfy the conditions:

$$w_j \geq 0 \text{ and } \sum_j w_j \leq 1 \text{ a.s.} \tag{3.4}$$

The weights are given by:

$$w_j = \lim_{n\to\infty} \frac{m_j(n)}{n} \text{ a.s.}, \tag{3.5}$$

which can be interpreted as the relative frequency of occurrence of the j-th observed species. Thus, they are determined by the prediction probability functions.

If the weights are such that $\sum_j w_j = 1$ almost surely, then the species sampling model is said to be proper and the random probability distribution reduces to:

$$F = \sum_{j=1}^{\infty} w_j \delta_{c_j}, \tag{3.6}$$

which is almost surely discrete. This is the case when $\lim_{n\to\infty} \frac{k(n)}{n} = 0$, almost surely.

F is the directing measure of the species sampling sequence and it is referred to as the *species sampling model* (SSM). We can then write our prior as:

$$\begin{aligned} F &\sim SSM(p_j, F_0) \\ z_1, \ldots, z_n \mid F &\overset{iid}{\sim} F, \end{aligned}$$

which states that conditional on F, the $z_i \in \left\{c_1, \ldots, c_{k(i-1)}, c_{k(i-1)+1}\right\}$ are i.i.d. draws from the random probability distribution. Marginalizing over F gives us the joint distribution, $p(Z)$, which is the product of the conditional distributions in (3.1). In practice, this is how we work with the SSM; without worrying about the specific form of F.

An important property of the species sampling model is the implied prior on the sequence of random partitions. As noted earlier, the sequence $(z_1, \ldots, z_n)$ implicitly defines a partition through the clustering of observations sharing the same label. Thus, the implied prior on the sequence of random partitions is equivalent to the joint

distribution of the indicator variables, $p(Z)$. As a sample from F, the z_i constitute a species sampling sequence and as such, are exchangeable with respect to their indices. This implies that the cluster sizes are sufficient to summarize the relevant information in Z so the prior can be written as $p(\mathbf{m}(n))$. Since the labels are generated based on the order of appearance of new species, any rearrangement of the variables may result in a differently ordered cluster size vector. It follows that the $p(\mathbf{m}(n))$ must be a symmetric function in order to ensure that any ordering of the cluster sizes is assigned the same probability. Such a function is known as an *exchangeable partition probability function* (EPPF) and can be defined for every species sampling sequence [143, 144]. Note that exchangeability of the z_i with respect to their labels follows from the fact that permuting the labels does not change the resulting partition (since the parts are defined by equality of the labels irrespective of their values), hence the cluster size vector $\mathbf{m}(n)$ remains unchanged. This confirms our earlier conclusion that $p(Z)$ should be partition exchangeable.

Formally, an EPPF must satisfy the following constraints in order to maintain consistency across samples sizes [101]:

$$p(\mathbf{m}(n)) = \sum_{j=1}^{k(n)+1} p\Big(\mathbf{m}(n)^{j+}\Big), \tag{3.7}$$

for all $\mathbf{m}(n)$, where $p(1) = 1$ and $\mathbf{m}(n)^{j+}$ denotes $\mathbf{m}(n)$ with the jth cluster size incremented by one. The prediction probability functions can be shown to be related to the EPPF as:

$$p_j(\mathbf{m}(n)) = \frac{p(\mathbf{m}(n)^{j+})}{p(\mathbf{m}(n))}, \tag{3.8}$$

for $j \in \{1, \ldots, k(n), k(n)+1\}$ provided $p(\mathbf{m}(n)) > 0$ [144].

At this point, it should be clear that species sampling models offer a compelling approach to the problem of how to define the prior distribution of the target cluster labels in the associative tracking model. For an unknown number of targets, we have seen that the proper interpretation of the indicator variables is as implicitly defining a partition of the observations. When viewed as a partition of the index set $\{1, \ldots, n\}$, each part contains the indices of the measurements assumed to have originated from a distinct target. Unfortunately, even knowing that $p(Z)$ should be defined on the space of partitions, it is by no means clear which parametric form the distribution should assume. Species sampling models provide a principled treatment of this uncertainty

by using a random probability distribution. Since we cannot reasonably assign prior probabilities to the targets we observe, since we did not know about them until they were observed, the species sampling model encodes a set of beliefs about the relative abundance of the observed targets. The EPPF provides the specific form for the desired prior.

3.3 Bayesian Nonparametric Priors

3.3.1 Dirichlet Process

The Dirichlet process (DP) [3, 53] is the most widely used Bayesian nonparametric prior due to both its simplicity and theoretical properties. It can be shown to be equivalent to a proper SSM with prediction probability functions given by [144]:

$$p_j(\mathbf{m}(n)) = \begin{cases} \frac{m_j(n)}{n+\alpha} & j \in \{1, \ldots, k(n)\} \\ \frac{\alpha}{n+\alpha} & j = k(n)+1 \end{cases}, \tag{3.9}$$

where $\alpha \in \mathbb{R}^+$ is a concentration parameter that controls the formation of new clusters.

As a proper SSM, it converges to a discrete random probability measure of the form (3.6) and we can write the DP prior as:

$$\begin{aligned} F &\sim DP(\alpha, F_0) \\ z_1, \ldots, z_n \mid F &\overset{iid}{\sim} F. \end{aligned}$$

The exchangeable partition probability function for the species sampling sequence is given by:

$$\begin{aligned} p(\mathbf{m}(n)) &= p(z_1, \ldots, z_n) \\ &= p(z_1)\, p(z_2 \mid z_1) \ldots p(z_n \mid z_1, \ldots, z_{n-1}) \\ &= \frac{\Gamma(\alpha)\, \alpha^{k(n)} \prod_{j=1}^{k(n)} \Gamma(m_j(n))}{\Gamma(n+\alpha)} \prod_{j=1}^{k(n)} F_0(c_j)\,. \end{aligned} \tag{3.10}$$

When we take F_0 to be a uniform distribution on the interval $[0, 1]$, the last product term is equal to one and can be dropped. In practice, we will simplify the model by marginalizing over F and work directly with (3.10).

The name of the process comes from the fact that for any finite measurable partition $(A_1, \ldots, A_p)$ of the space on which F is defined, the random vector $(F(A_1), \ldots, F(A_p))$ is distributed as a finite-dimensional Dirichlet distribution [185]:

$$(F(A_1), \ldots, F(A_p)) \sim Dir(\alpha F_0(A_1), \ldots, \alpha F_0(A_p)).$$

We can view the DP as sequentially constructing a partition of Z. The first indicator variable, z_1, is assigned a new label because $k(n) = 0$ and $n = 0$. At each step, the i-th indicator variable is assigned to one of the existing labels with a probability that is proportional to the size of the part, or assigned a new label with a probability that is proportional to α. This implies that the DP prior exhibits a "rich-get-richer" behavior when it comes to the cluster sizes.

The same partition structure that is implied by the DP can also be derived in different ways. The first uses the generalized Pólya urn scheme of Blackwell and MacQueen [14]. It imagines an urn containing balls of distinct colors. The balls correspond to the indicator variables, z_i, and the colors correspond to the labels, c_j. When a ball is drawn, it is placed back in the urn together with another ball of the same color. With probability proportional to α, a new c_j is drawn from F_0 and a ball of a new color is added to the urn. A second way to derive the partition structure equivalent to (3.10) is by using the Chinese restaurant process from Dubins and Pitman [144]. It is based on the metaphor of a Chinese restaurant with an unlimited number of tables, each capable of seating an unlimited number of customers. Here, the customers correspond to the indicator variables, z_i, and the tables correspond to the labels, c_j. As customers enter the restaurant one at a time, they choose a table to sit down at with a probability proportional to the number of customers already seated there. With probability proportional to α, the customer sits at a new table, for which a new label c_j is drawn from F_0. The DP partition structure can also be derived using the Ewens sampling formula [51, 3] from population genetics.

It is the discrete nature of the DP that makes it well-suited for use as a prior in clustering applications because there is a positive probability of ties among the z_i. In the context of SSMs, we assumed that we were drawing the arbitrary labels used to identify the observed species. However, this not the only way to exploit the discrete nature of the DP for clustering. It is perhaps even more common to see a model such

as:

$$\begin{aligned} F &\sim DP(\alpha, F_0) \\ \theta_1, \ldots, \theta_n \mid F &\overset{iid}{\sim} F. \\ y_i \mid \theta_i &\sim F_Y(\theta_i)\,, \end{aligned}$$

where F_0 is a distribution over parameters (rather than labels) and each measurement, y_i, is assumed to be probabilistically generated from the latent parameter, $\theta_i \in \left\{c_1, \ldots, c_{k(i-1)}, c_{k(i-1)+1}\right\}$, that is drawn from the DP. In this context, it is the ties among the parameter values that implicitly represent the partition structure, hence the clusters. Such a model is known as a DP mixture model because it is equivalent to a mixture with a countably infinite number of components of the form:

$$p(y_i \mid \theta_i) = \sum_{j \geq 1} \pi_j p(y_i \mid \theta_i = c_j)\,,$$

where the π_j are the mixture weights, which are given by the stick-breaking representation of the DP [174].

Citing various criticisms of clustering with mixture models, Casella et al. [26] argue that a better approach is to use product partition models [80]. Product partition models are of the form:

$$p(Y \mid Z) = \prod_{j=1}^{K} p\Big(Y_{c_j} \mid c_j\Big)\,,$$

where $Y = \{y_1, \ldots, y_n\}$ is the set of measurements and Y_{c_j} denotes the subset containing those measurements assigned to the j-th (of K) part. The measurements in Y_{c_j} are assumed to be probabilistically generated from the parameter c_j. However, when the mixture model is based on a nonparametric model such as the DP, the two approaches are in fact related. If we marginalize over F, as is the usual case in practice, the partition structure is retained and the model reduces to that the product partition model [157]. Note that the associative tracking model is based on the product partition model. However, we could have arrived at the same result by starting with a DP mixture model, choosing a base distribution that is conjugate to the likelihood, and then marginalizing over both the parameters and F. This is exactly what was done by Fox et al. [60] in their target tracking algorithm.

There is also a two-parameter generalization of the DP known as the Pitman-Yor process [145, 191]. It too is a proper SSM with prediction probability functions

given by:

$$p(z_{n+1} = c_j \mid z_1, z_2, \ldots, z_n) = \begin{cases} \frac{m_j(n)-\beta}{n+\alpha} & j \in \{1, \ldots, k(n)\} \\ \frac{\alpha+k(n)\beta}{n+\alpha} & j = k(n)+1 \end{cases}, \tag{3.11}$$

where β is a discount parameter that reduces the probability of adding the current observation to an existing cluster. To ensure valid probabilities, the parameters are chosen such that $0 \leq \beta < 1$ and $\alpha > -\beta$, where $\beta = 0$ reduces the model to the DP. The corresponding exchangeable partition probability function is:

$$p(z_1, \ldots, z_n) = \frac{\left(\prod_{j=1}^{k(n)-1} (\alpha + j\beta)\right) \left(\prod_{j=1}^{k(n)} [1-\beta]_{m_j(n)-1}\right)}{[1+\beta]_{n-1}} \prod_{j=1}^{k(n)} F_0(c_j), \tag{3.12}$$

where $[x]_m = \prod_{j=1}^{m} (x + j - 1)$. When $0 < \beta < 1$, the cluster sizes exhibit power-law behavior which is important for certain applications. However, we will not consider the Pitman-Yor process further here.

For a given concentration parameter, α, and number of measurements, n, the distribution of the number of clusters produced by the DP may be written as [3]:

$$p(k \mid \alpha, n) = S_1(n, k)\, \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)}, \tag{3.13}$$

where $k \in \{1, \ldots, n\}$ is the number of clusters and $S_1(n, k)$ is the unsigned Stirling number of the first kind. These Stirling numbers count the number of permutations of n elements with k disjoint cycles. They can be calculated using the recurrence relation:

$$S_1(n+1, k) = nS_1(n, k) + S_1(n, k-1),$$

with $S_1(0, 0) = 1$ and $S_1(n, 0) = S_1(0, n) = 0$, for $n > 0$ and $k > 0$. However, evaluating (3.13) can be difficult in practice because the values of $S_1(n, k)$ and $\Gamma(\alpha + n)$ can quickly grow too large for most numerical representations. See appendix B for details on numerically stable method of evaluation.

The distribution over k is used to specify the hyperprior for the concentration parameter. Recall that in the last chapter we explained in detail how this was done by matching $p(k \mid \alpha, n)$ to an expression of our prior beliefs about the number of clusters.

The mean and variance of k have closed-form expressions given by:

$$\begin{aligned} E[k \mid \alpha, n] &= \sum_{i=1}^{n} \frac{\alpha}{\alpha + i - 1} \\ &= \alpha \left[\psi_0(\alpha + n) - \psi_0(\alpha)\right] \end{aligned}$$

and

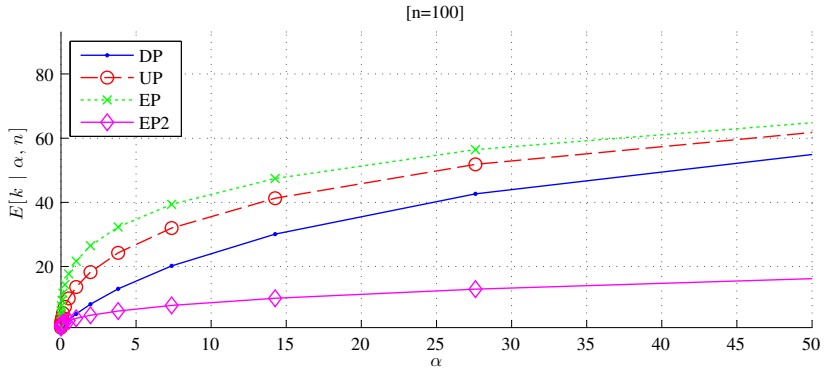
$$Var[k \mid \alpha, n] = \sum_{i=1}^{n} \frac{\alpha\,(i-1)}{(\alpha+i-1)^2}$$
$$= \alpha\left[\psi_0(\alpha+n) - \psi_0(\alpha)\right] + \alpha^2\left[\psi_1(\alpha+n) - \psi_1(\alpha)\right],$$

where $\psi_0(\cdot)$ denotes the digamma function and $\psi_1(\cdot)$ denotes the trigamma function.

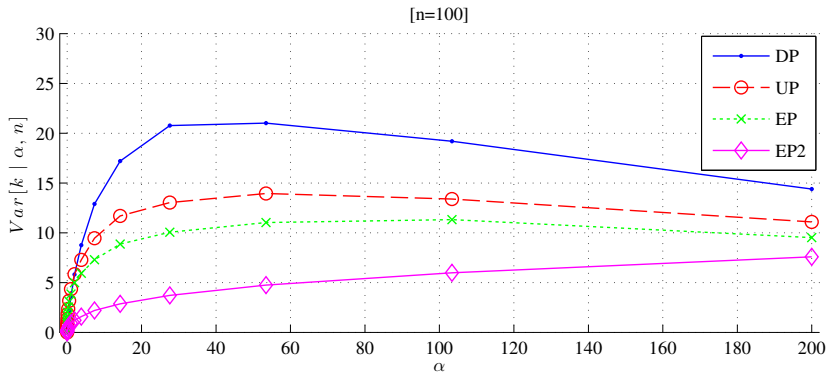
Figure 3.1 shows how the mean and variance of k vary as a function of α (the other nonparametric prior processes are included for comparison). We can see that as $\alpha \to 0$, the expected number of clusters goes to one and the DP tends to avoid creating any new clusters in favor of adding every measurement to the initial cluster. As $\alpha \to \infty$, the expected number of clusters goes to n and the DP tends toward placing every measurement in its own cluster. Note that the $E[k \mid \alpha, n]$ plot has been truncated to $\alpha = \frac{n}{2}$ to focus on the more interesting behavior for small values of the concentration parameter. The variance in the number of clusters reaches its peak just before $\alpha = \frac{n}{2}$ so the $Var[k \mid \alpha, n]$ plot has been extended to $\alpha = 2n$ to show more of its shape. Beyond these limits the mean approaches the asymptote at n and the variance decays to zero.

As the prediction probability functions suggest, the DP induces an unequal distribution of cluster sizes with a few clusters receiving most of the measurements. Figure 3.2 shows the distribution of clusters sizes for 25000 realizations of the DP with α set to its most likely value based on the hyperprior for the given η, *minClusterSz*, and n. Indeed, as the cluster size increases the number of clusters of that size decreases exponentially. The DP produces more singleton clusters than anything else.

Despite the theoretical elegance of the DP and its ability to model our uncertainty about the prior distribution of the target cluster labels using a random probability distribution, there is no free lunch. As the figure 3.2 makes clear, the DP entails some strong assumptions about the partition structure that may not be consistent with our prior beliefs. For example, do we really have a strong preference for the partition $\mathbf{m}(100) = [99, 1]$ over $\mathbf{m}(100) = [75, 25]$ as the DP would suggest? It would seem unlikely. In the next section, we will look at another stochastic process in an attempt to retain the theoretical elegance of the SSM but with assumptions about the partition structure that are perhaps better suited to tracking applications.



(a) Mean cluster counts.



(b) Variance of the cluster counts.

Figure 3.1: Mean and variance, as a function of α, of the number of clusters produced by the nonparametric prior processes.

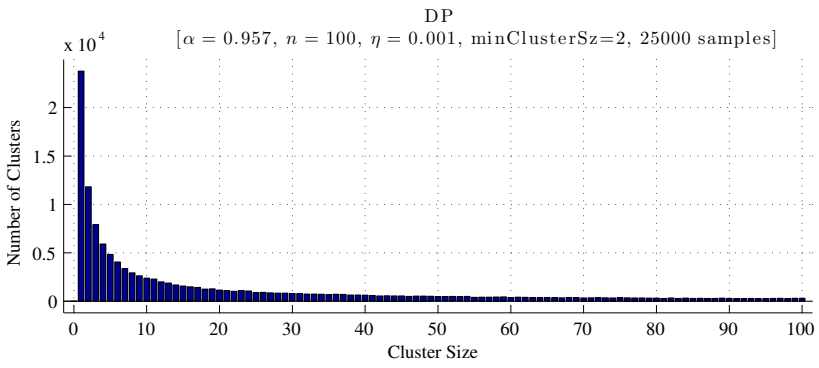


Figure 3.2: Distribution of clusters sizes for the Dirichlet process.

3.3.2 Uniform Process

Despite its ubiquity, the DP is not well-suited as a prior distribution for all clustering applications because of its preference for unequal cluster sizes. Its widespread use is due in large part to the fact that the DP prior leads to tractable inference algorithms, however, it has enjoyed notable success when applied to infinite mixture models. This should come as no surprise because, with respect to the mixture density, it is often reasonable to have a few components dominate while most have negligible weights. On the other hand, when our interest lies the partition structure itself, that may very well be a problem. For tracking applications, it implies some rather unlikely beliefs such as the prevalence of small clusters.

Since the partition structure induced by the DP is not a particularly realistic model for target cluster sizes, as an alternative we will consider the uniform process (UP) [155, 89]. The UP was first introduced as a prior for DNA motif clustering and has seen surprising little interest since. It was motivated by the desire to have nonparametric prior that induces a partition structure with a more uniform distribution of cluster sizes. Unfortunately, abandoning the unequal clustering produced by the DP (and the Pitman-Yor process) comes at the expense of exchangeability. Thus, the UP is not a true SSM, but in practice it can be made to approximate one to an acceptable degree.

Its prediction probability functions are given by:

$$p_j(\mathbf{m}(n)) = \begin{cases} \frac{1}{k(n)+\alpha} & j \in \{1, \ldots, k(n)\} \\ \frac{\alpha}{k(n)+\alpha} & j = k(n)+1, \end{cases} \tag{3.14}$$

where $\alpha \in \mathbb{R}^+$ is a concentration parameter that controls the formation of new clusters.

From the perspective of sequentially constructing a partition of Z, we can see that the UP aims to induce a uniform distribution of cluster sizes by assigning each indicator variable one of the existing labels with equal probability. However, despite the fact that these prediction probability functions are functions of the cluster sizes, $\mathbf{m}(n)$, the sequence described by the prediction rule based on (3.14) is not exchangeable. This is because each $p_j(\mathbf{m}(n))$ depends on the previous assignments through the number of clusters. Consequently, any $(z_1, z_2, \ldots)$ generated by the UP is not a species sampling sequence.

Wallach et al. [191] studied the UP and concluded that the lack of exchange-

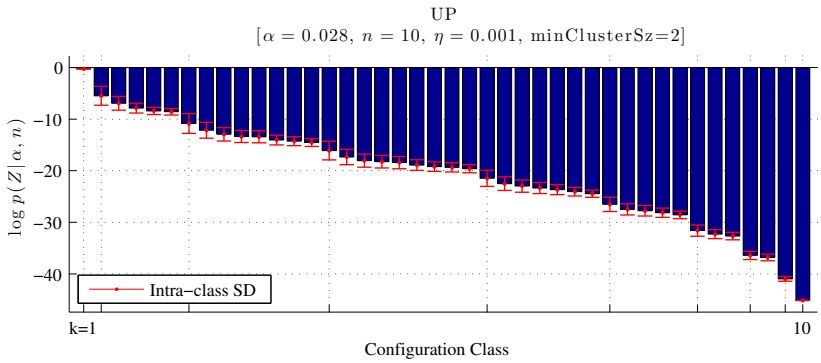


Figure 3.3: Variability of $p(Z)$ with respect to the order of the indicator variables for the uniform process.

ability is not a significant problem in practice and that $p(Z)$ can be highly robust to different orderings of the indicators. Specifically, they compared the standard deviation of $\log p(Z)$ for different orderings of the indicators within the same configuration class to the standard deviation of $\log p(Z)$ between configuration classes, noting that the latter was considerably larger. A configuration class is the class of partitions with equivalent cluster size vectors. In other words, all Z for which the corresponding $\mathbf{m}(n)$ have the same distribution of sizes are in the same configuration class. Figure 3.3 illustrates this notion of the sensitivity of $p(Z)$ to different orderings using a small example, $n = 10$, for which we can explicitly enumerate and readily visualize all the configuration classes. The error bars indicate the standard deviation of $\log p(Z)$ for a given configuration class; α is set to its most likely value based on the hyperprior. In this case, the average standard deviation within configuration classes is 1.03 while the standard deviation between configuration classes is 9.9.

To study this behavior for larger and more realistic values of n, we have to rely on simulation. This is done by simulating the UP, for a given n and α set to its most likely value based on the hyperprior, and then keeping track of the unique (after sorting) cluster size vectors that are produced. We take the results to be a reasonable sample of configuration classes over which we can compute the variability of $p(Z)$ for random orderings. For each value of n evaluated, we compute the standard deviation of $\log p(Z)$ both within configuration classes and between configuration classes. The former is computed by averaging the intra-class standard deviations over the sample of classes. The latter is computed as the standard deviation of the mean values for each

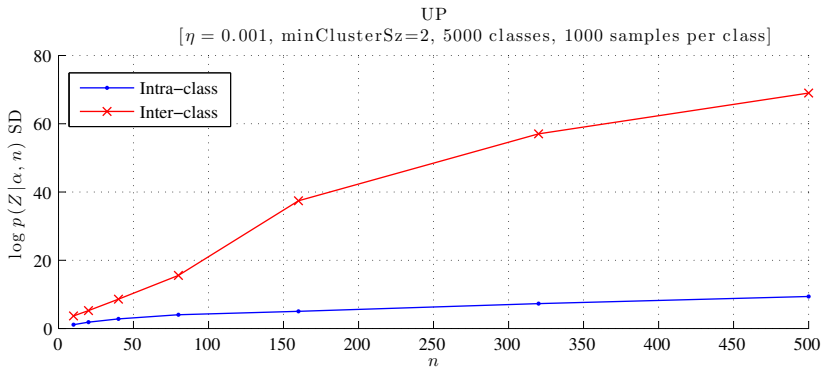


Figure 3.4: Comparison of how $p(Z)$ varies, within configuration classes and between configuration classes, with respect to the order of the indicator variables for the uniform process.

class in the sample. Figure 3.4 shows how the two standard deviations compare for $10 \leq n \leq 500$. The variability between configuration classes is seen to be considerably larger than the variability with classes, with the gap growing approximately linearly with n.

Although the UP is close, in some sense, to an exchangeable data generating process, an important consequence of the fact that it is not an SSM is that we are not guaranteed the existence of an EPPF. However, an approximated EPPF can be derived by using a "signature" of the cluster size vector to ensure that any arrangement of the indicator variables, hence any ordering of the cluster sizes, is assigned the same probability. For example, Jensen and Liu [89] suggest sorting it from the largest cluster to the smallest. Let $\mathbf{m}'(n)$ denote the cluster size vector sorted in descending order. The approximated EPPF can then be written as:

$$\begin{aligned} p(\mathbf{m}'(n)) &\approx p(z_1, \ldots, z_n) \\ &\approx p(z_1)\, p(z_2 \mid z_1) \ldots p(z_n \mid z_1, \ldots, z_{n-1}) \\ &= \alpha^{k(n)-1} \left(\alpha + k(n)\right) \prod_{i=1}^{k(n)} (\alpha + i)^{-m'_i(n)} . \end{aligned} \tag{3.15}$$

Of course, this approximated EPPF is based on processing the measurements in a particular order. It is equivalent to sequentially constructing the partition by adding all of the measurements assigned to the first cluster, then all of the measurements assigned to the next cluster, and so on; with the order in which the clusters are processed based on the signature. This is a valid approximation to an EPPF but it is not the

only possibility, even for the same signature. Ideally, we would like to use the average $p(Z)$ taken over all arrangements of the indicators for a given $\mathbf{m}'(n)$, but to compute this average exactly would be generally intractable. However, we can approximate it to some extent in order to improve our EPPF. The basic idea is to combine low and high estimates based on processing the measurements in different orders. It turns out that (3.15) yields a high estimate of $p(Z)$ for a given $\mathbf{m}'(n)$. We can compute a low estimate by sequentially constructing the partition by first adding one measurement to each cluster and then returning to the cluster-at-a-time approach for the remaining measurements. This can be written as:

$$p(\mathbf{m}'(n)) = \frac{\alpha^{k(n)}\Gamma(\alpha)}{\Gamma(\alpha + k(n))\,(\alpha + k(n))^{\sum_i m_i(n) - k(n)}}. \tag{3.16}$$

By averaging (3.15) and (3.16), in log-space, we get an improved EPPF that is closer to the ideal construction. We call this the averaged EPPF.

Since the EPPF is crucial to our model, it is important that we gauge the accuracy of the two approximations. A practical way to do that is to look at the impact of the EPPF on the distribution of the number of clusters. This makes particular sense for our model because the hyperprior for α is defined such that it depends on this distribution.

For small values of n, we can compute $p(k \mid \alpha,\, n)$ using the EPPF by explicitly enumerating all possible cluster size vectors. Note that for a given cluster size vector, we have to scale the probability by the number of unique arrangements of the indicator variables that give rise to it in order to correctly compute $p(k \mid \alpha,\, n)$. Those distributions are then compared to one obtained by simulating the UP. Figure 3.5 shows such a comparison for $n = 15$ and α set to its most likely value based on the hyperprior. The averaged EPPF does indeed yield a modest improvement. In this case, the number of measurements is small enough that we can readily visualize $\log p(Z)$, for each configuration class, to compare the averaged EPPF to the ideal values obtained via simulation. Figure 3.6 shows just how similar the resulting probabilities are. Note that here the $\log p(Z)$ values have each been scaled by the number of arrangements of the indicators that will yield the equivalent configuration class so that the distribution displayed is properly normalized with respect to Z.

To see what happens for larger values of n, we can no longer compute $p(k \mid \alpha,\, n)$ via explicit enumeration. Instead, we can simulate the UP for a large number of

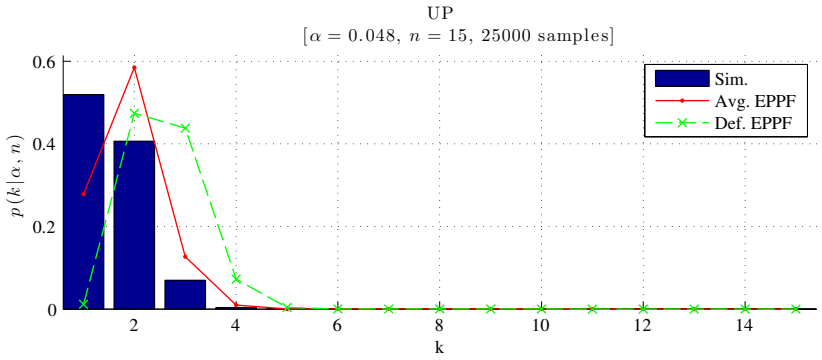


Figure 3.5: Comparing the accuracy of the default and averaged EPPFs by looking at the resulting distribution over the number of clusters for the uniform process.

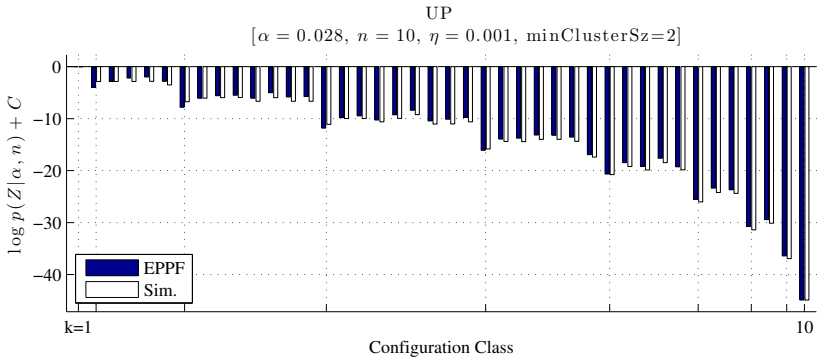


Figure 3.6: Comparison of the averaged EPPF probabilities to the simulated true values for the uniform process.

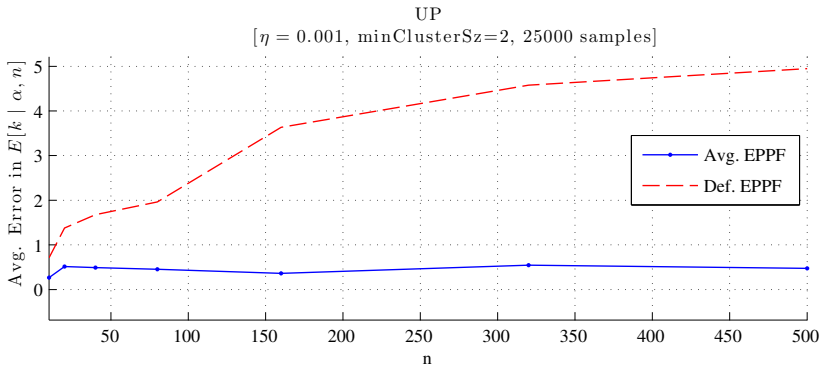


Figure 3.7: EPPF accuracy for the uniform process.

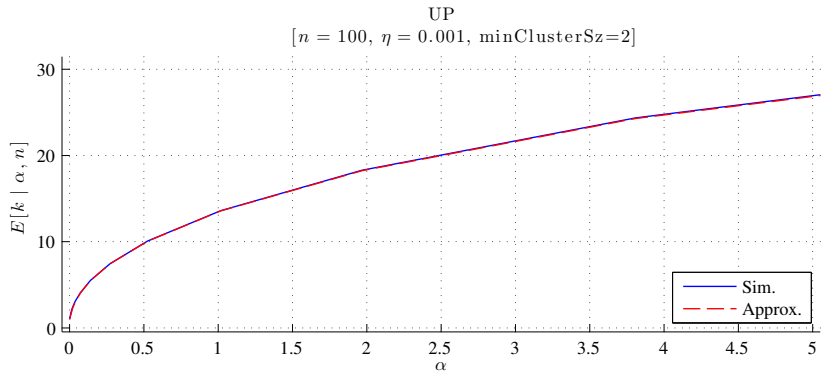
realizations and keep track of the unique cluster size vectors that are produced. We take these to be a reasonable sample from which we can estimate the expected the number of clusters using the EPPF. This value is then compared to the expected number of clusters obtained from the simulations. For each value of n evaluated, we set α to its most likely value based on the hyperprior. Using this procedure to examine the errors for $10 \leq n \leq 500$, we find that the original EPPF (equation (3.15)) increasingly overestimates the expected number of clusters as the number of measurements grows. By comparison, the averaged EPPF (equation (3.16)) overestimates more or less consistently by about 0.45 for all n. The results are shown in figure 3.7. This confirms the advantage of using the averaged EPPF and gives us confidence that it is sufficiently accurate to use for our model.

There is no closed-form expression for the distribution of the number of clusters produced by the UP. However, we can approximate it reasonably well using the moments, $E[k \mid \alpha, n]$ and $Var[k \mid \alpha, n]$. First, let us look at how to compute the moments.

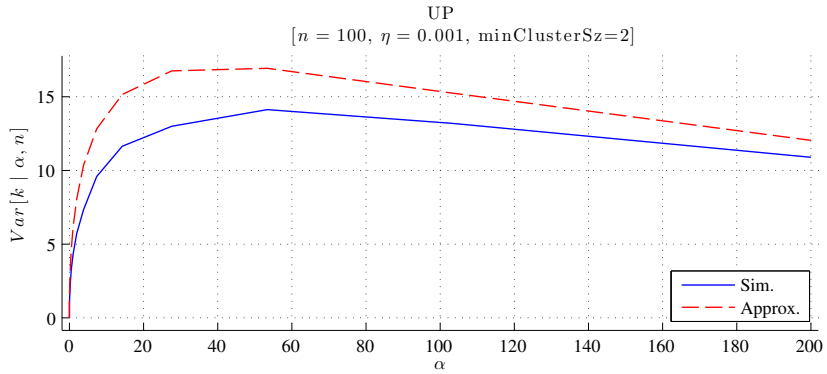
Using the number of observations needed until the k-th new cluster is generated, which is explained in the supplementary materials for [191], we can derive a closed-form estimate for the mean number of clusters, which is given by:

$$E[k \mid \alpha, n] \approx \frac{1}{2}(1 - 2\alpha + \sqrt{1 - 4\alpha + 8N\alpha + 4\alpha^2}).$$

Based on evaluations using UP simulations, this estimate is accurate to less than 0.2. The variance in the number of clusters can also be estimated in closed-form, albeit with



(a)



(b)

Figure 3.8: Accuracy of the estimated mean and variance of the number of clusters for the uniform process.

less accuracy. That estimate is given by:

$$Var[k \mid \alpha, n] \approx \frac{1}{2}\alpha\left(-2 + \frac{-4 + 8N + 8\alpha}{2\sqrt{1 - 4\alpha + 8N\alpha + 4\alpha^2}}\right).$$

It it overestimates the variance by about 10–15% but more importantly conforms to correct shape as a function of α. Figure 3.8 shows an example of how these estimates compare to the values obtained by simulation. Referring back to figure 3.1, we can see how the mean and variance of the number of clusters produced by the UP, as a function of α, compares to the other nonparametric processes. The limiting behavior is the same for all processes, as we would expect. The rate of growth in $E[k \mid \alpha, n]$ for UP, however, is faster than that of the DP and it has a lower variance for a given α.

Given estimates for the mean and variance of k, we can approximate the distribution of the number of clusters. Let e and v be the mean and variance of the number

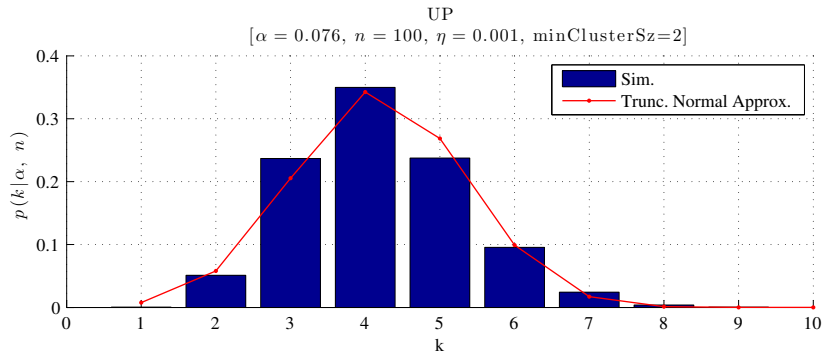


Figure 3.9: Example of an approximation to $p(k \mid \alpha, n)$, computed from its first two moments, for the uniform process.

of clusters, respectively. The distribution can then be approximated using a truncated Gaussian distribution:

$$p\left(k \mid \alpha, n\right) \propto \exp\left(\frac{-\left(k-e\right)^2}{2v}\right),$$

where $k \in [1..n]$. Figure 3.9 shows an example of such an approximation. For small values of α, a slightly better estimate (in terms of mean squared error compared to the simulated distribution) can be obtained using a truncated gamma distribution:

$$p\left(k \mid \cdot\right) \propto k^{a-1} \exp\left(-bk\right),$$

where $a = \frac{e^2}{v}$, $b = \frac{e}{v}$, and $k \in [1..n]$. Note that these approximations can also be used for the DP as well.

As the prediction probability functions suggest, the UP induces an equal distribution of cluster sizes. Figure 3.10 shows the distribution of clusters sizes for 25000 realizations of the UP with α set to its most likely value based on the hyperprior for the given η, *minClusterSz*, and n. The breakdown in the uniformity that begins around a cluster size of 25 occurs because of the interplay between α and the finite number of measurements. For the given α, the expected number of clusters is 4.13, which implies clusters of size 24.2. Beyond this, the process produces exponentially fewer clusters.

3.3.3 Egalitarian Process

The UP provides an alternative to the "rich-get-richer" behavior of the DP by favoring a more balanced distribution of cluster sizes. From the perspective of sequentially constructing the partition, it gives each existing cluster an equal chance

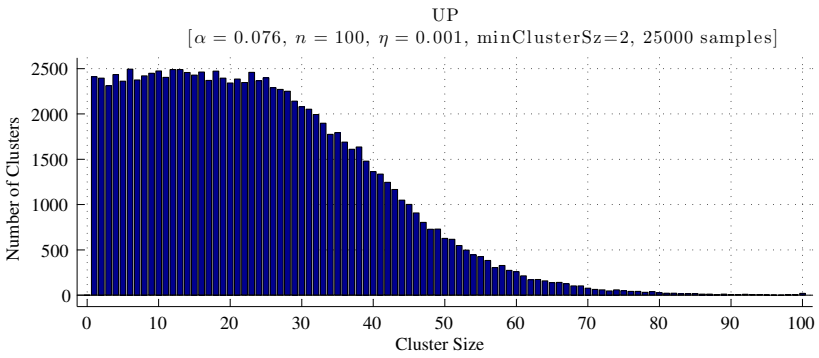


Figure 3.10: Distribution of clusters sizes produced by the uniform process.

to take the next measurement by ignoring their sizes and not favoring any one of them. However, in tracking applications we often want to express a preference for large clusters (i.e. long tracks). One way to do this would be to actually favor smaller clusters when assigning a new measurement.

No well-known stochastic process with these properties exists, but we can create one using on the prediction probability functions:

$$p_j(\mathbf{m}(n)) \propto \begin{cases} \frac{1}{m_j(n)} & j \in \{1, \ldots, k(n)\} \\ \frac{1}{\alpha} & j = k(n)+1 \end{cases}, \tag{3.17}$$

where $\alpha \in \mathbb{R}^+$ is a concentration parameter that controls the formation of new clusters. Note that these are proportional probabilities; the $p_j(\mathbf{m}(n))$ for $j \in \{1, \ldots, k(n)\}$ must all be known in order to compute the normalizing constant.

For lack of a better term, we call this the egalitarian process (EP) because it actively strives to maintain equality of the cluster sizes by assigning each indicator variable one of the existing labels with a probability that is inversely proportional to the size of the cluster. Like the UP, the EP lacks exchangeability because each $p_j(\mathbf{m}(n))$ depends on the previous assignments; this time through the cluster sizes. Thus, it too fails to satisfy the conditions necessary to be an SSM.

To see how robust $p(Z)$ is to different orderings of the indicators for the EP, we start by looking at the same small example of the variability of $p(Z)$ for random orderings that we considered for the UP. See figure 3.11. The variability within configuration classes has clearly increased over what we observed in 3.3 for the UP but remains considerably smaller than the variability between configuration classes. For larger values

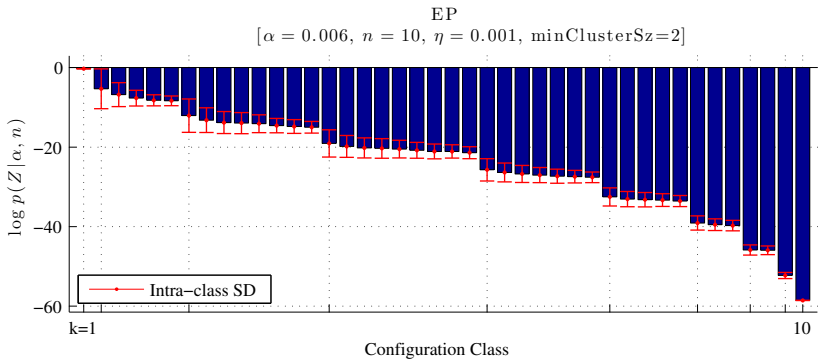


Figure 3.11: Variability of $p(Z)$ with respect to the order of the indicator variables for the egalitarian process.

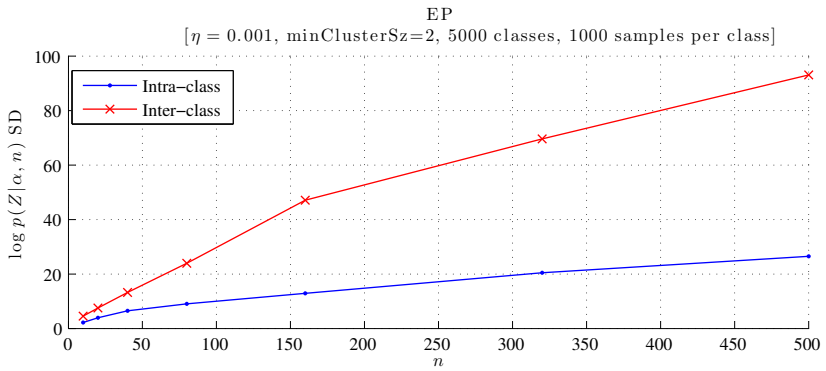


Figure 3.12: Comparison of how $p(Z)$ varies, within configuration classes and between configuration classes, with respect to the order of the indicator variables for the egalitarian process.

of n, figure 3.12 shows that the two standard deviations follow a similar trend to that of the UP (figure 3.4) but with almost twice intra-class variability.

It is perhaps unsurprising that the EP is more sensitive to different orderings of the indicators than the UP. This is presumably because each of the EP prediction probability functions has a stronger dependence on the previous assignments. They depend on the previous assignments not only through the number of clusters but through their individual sizes as well. A obvious question then arises: can we express a similar preference for assigning new measurements to smaller clusters but with a weaker dependence on the past?

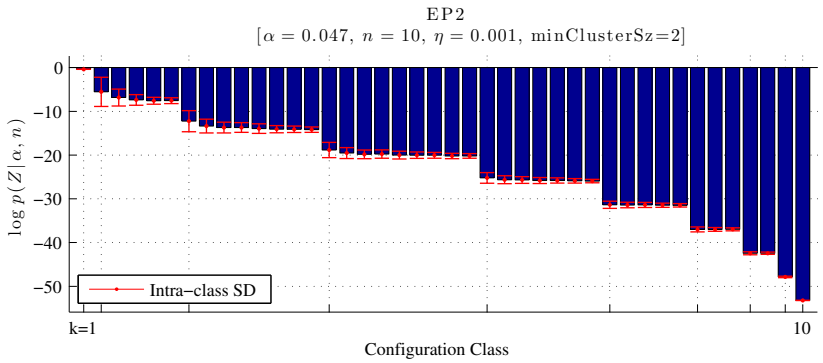


Figure 3.13: Variability of $p(Z)$ with respect to the order of the indicator variables for the second form of the egalitarian process.

One way to do this is using the prediction probability functions:

$$p_j(\mathbf{m}(n)) = \begin{cases} \frac{n+1-m_j(n)}{k(n)(n+1)-n+\alpha} & j \in \{1, \ldots, k(n)\} \\ \frac{\alpha}{k(n)(n+1)-n+\alpha} & j = k(n)+1. \end{cases} \tag{3.18}$$

These are based on the notion that the probability of selecting one of the existing clusters is proportional to the difference between its current size and its maximum possible size, assuming no knowledge of the other cluster sizes. This second form the EP is denoted by EP2.

For the small example of the variability of $p(Z)$ for random orderings, figure 3.13 shows a marked reduction in the variability within configuration classes, compared to the EP, with similar variability between configuration classes. In fact, the average standard deviation within configuration classes in this case is only 0.97, which is lower than that of the UP. For larger values of n, figure 3.14 shows that the variability between configuration classes is again considerably larger than the variability within classes, and with the gap growing approximately linearly with n. Comparing that to figures 3.4 and 3.12, the EP2 clearly exhibits the least sensitivity to different orderings of the indicators, with significantly lower intra-class variability than either the UP or the EP.

Approximated EPPFs for both the EP and EP2 can be derived in the same manner as was done for the UP, using a signature of the cluster size vector. In fact, we use the same signature: the cluster size vector sorted in descending order. The

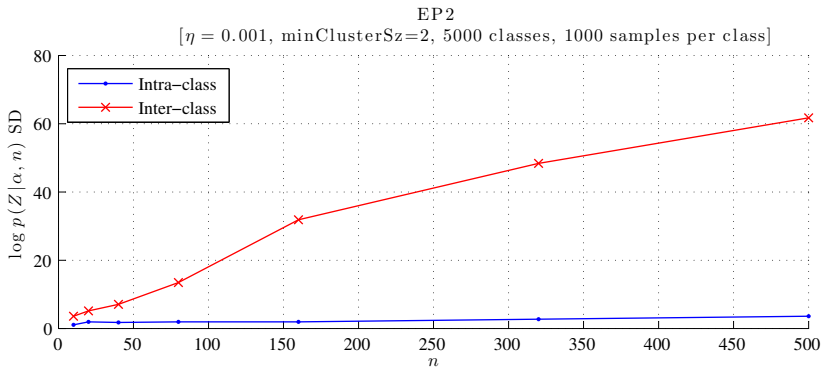


Figure 3.14: Comparison of how $p(Z)$ varies, within configuration classes and between configuration classes, with respect to the order of the indicator variables for the second form of the egalitarian process.

approximated EPPF for the EP can then be written as:

$$\begin{aligned} p(\mathbf{m}'(n)) &\approx p(z_1, \ldots, z_n) \\ &\approx p(z_1)\, p(z_2 \mid z_1) \ldots p(z_n \mid z_1, \ldots, z_{n-1}) \\ &= \prod_{i=1}^{k(n)} \left[\frac{\alpha}{\sum_{l=1}^{i-1} 1/m'_l(n) + \alpha} \prod_{j=1}^{m'_i(n)-1} \frac{(1/j)}{\sum_{l=1}^{i-1} 1/m'_l(n) + (1/j) + \alpha} \right]. \end{aligned} \tag{3.19}$$

This is based on processing the measurements such that each cluster is added one after the other, with the order in which the clusters are processed determined by the signature. As was the case for the UP, this tends to produce a high estimate of $p(Z)$ for a given $\mathbf{m}'(n)$. We can again compute a low estimate by sequentially constructing the partition by first adding one measurement to each cluster and then returning to the cluster-at-a-time approach for the remaining measurements. This can be written as:

$$p(\mathbf{m}'(n)) = \frac{\alpha^{k(n)}\Gamma(\alpha)}{\Gamma(k(n)+\alpha)} \prod_{i=1}^{k(n)} \left[\prod_{j=1}^{m'_i(n)-1} \frac{(1/j)}{\sum_{l=1}^{i-1} 1/m'_l(n) + (k(n)-i) + (1/j) + \alpha} \right]. \tag{3.20}$$

As before, we average the two in log-space to improve the approximation.

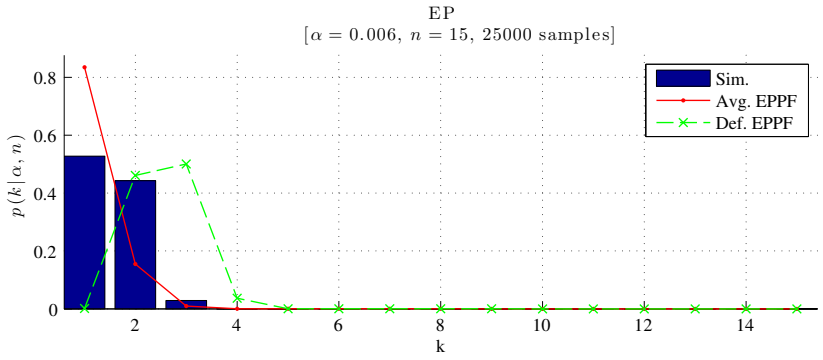


Figure 3.15: Comparing the accuracy of the default and averaged EPPFs by looking at the resulting distribution over the number of clusters for the egalitarian process.

The approximated EPPF for the EP2 can be written as:

$$
\begin{aligned}
p(\mathbf{m}'(n)) &\approx p(z_1, \ldots, z_n) \\
&\approx p(z_1)\, p(z_2 \mid z_1) \ldots p(z_n \mid z_1, \ldots, z_{n-1}) \\
&= \prod_{i=1}^{k(n)} \left[\frac{\alpha}{\left(\sum_{l=1}^{i-1} m'_l(n)\right)(i-2) + i - 1 + \alpha} \right. \\
&\qquad \left. \prod_{j=1}^{m'_i(n)} \frac{\left(\sum_{l=1}^{i-1} m'_l(n) + j\right) - m'_j(n)}{\left(\sum_{l=1}^{i-1} m'_l(n) + j - 1\right)(i-2) + i - 1 + \alpha} \right],
\end{aligned}
$$

which gives the high estimate of $p(Z)$. The corresponding low estimate is given by:

$$
p(\mathbf{m}'(n)) = \frac{\alpha^{k(n)}\Gamma(\alpha)}{\Gamma(k(n)+\alpha)} \prod_{i=1}^{k(n)} \left[\prod_{j=2}^{m'_i(n)} \frac{\left(\sum_{l=1}^{i-1} m'_l(n) + j\right) - m'_j(n)}{\left(\sum_{l=1}^{i-1} m'_l(n) + j - 1\right)(k(n)-1) + k(n) + \alpha} \right].
$$

To gauge the accuracy of the EPPF approximations, we again look at the impact of the EPPF on the distribution of the number of clusters. We begin by considering $p(k \mid \alpha, n)$ computed using the EPPF by explicitly enumerating all possible cluster size vectors. Figures 3.15 and 3.16 show the results for $n = 15$ and α set to its most likely value based on the hyperprior. In both cases, the averaged EPPF yields a distribution of cluster sizes that is closer to the simulated ideal.

For larger values of n, we use the same simulation-based procedure that we employed to gauge the accuracy of the EPPF for the UP. Figures 3.17 and 3.18 show the results for $10 \leq n \leq 500$. The advantage of the averaged EPPF is again confirmed. For both processes, the EPPF overestimates the expected number of clusters for all n. As

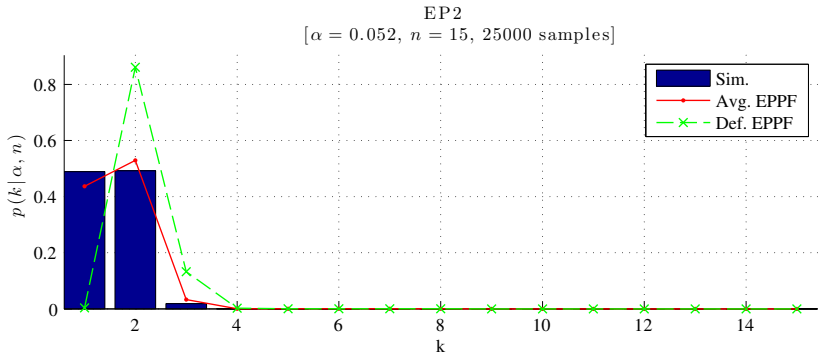


Figure 3.16: Comparing the accuracy of the default and averaged EPPFs by looking at the resulting distribution over the number of clusters for the second form of the egalitarian process.

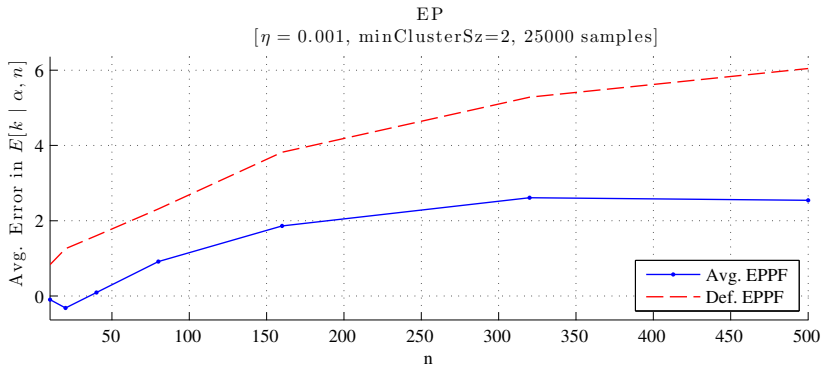


Figure 3.17: EPPF accuracy for the egalitarian process.

the number of measurements increases, the error increases but then starts to plateau. Repeated evaluations show some variation in how fast the error levels off, but in all evaluations maximum error remain just above 2.0. For the EP, it is not surprising to see that the error is larger than what we observed for the UP (figure 3.7). However, what is unexpected is that the EP2, given its significantly lower intra-class variability of $p(Z)$, only performs marginally better than the EP. This is because in the case of the EP2, despite the lower variability, the shape of the distribution of the $p(Z)$ values for the different orderings deviates even further from what is implied by the averaging, in log-space, of the high and low estimates. Additional research will be needed to determine if simple means of improving the EPPF approximation can be found.

There is no closed-form expression for the distribution of the number of clusters

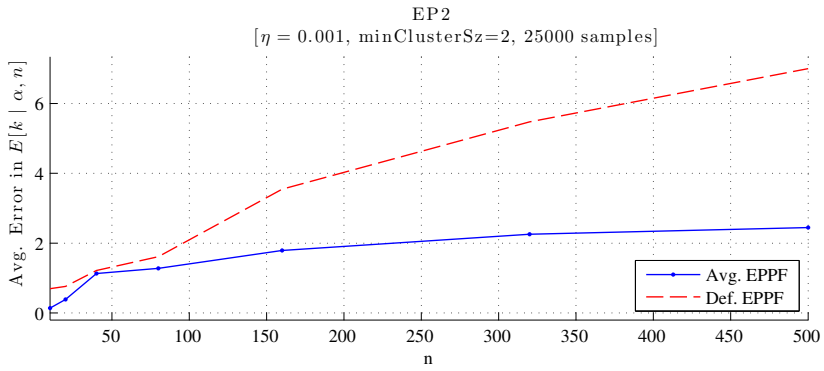


Figure 3.18: EPPF accuracy for the second form of the egalitarian process.

produced by either the EP or EP2 so it must be approximated using the moments, $E[k \mid \alpha, n]$ and $Var[k \mid \alpha, n]$, as was the case for the UP. However, for these processes we lack a closed-form estimate for the moments as well. Instead, we can derive an estimate procedurally. The basic idea is to model the process of sequentially constructing the partition in order to estimate the probability that each indicator variable is assigned to a new cluster. The procedure is as follows:

0. Preliminaries

   (a) Let $\mathbf{a} \in \mathbb{R}^n$ denote the "abundances" such that a_i is the number of clusters of size i, for $i \in [1..n]$.

   (b) Let e denote the estimate of $E[k \mid \alpha, n]$.

   (c) Let v denote the estimate of $Var[k \mid \alpha, n]$.

   (d) Let $\mathbf{p} \in \mathbb{R}^{n+1}$ denote the assignment probabilities such that p_i is the probability the current measurement is assigned to a cluster of size i, for $i \in [1..n]$, or a new cluster is created for $i = n + 1$.

1. Initialization ($i = 1$)

   (a) Set the abundance vector to one cluster of size one: $\mathbf{a} = [1\ 0\ \cdots\ 0]$.

   (b) Set the expected number of clusters after processing the first measurement to one: $e = 1$.

   (c) Set the variance of the number of clusters after processing the first measurement to zero: $v = 0$.

2. For $i = 2..n$

   (a) Compute $\mathbf{p}$: $p_j = \frac{a_j}{j}$ for the EP or $p_j = a_j\,(i-j)$ for the EP2, for $1 \le j \le n$, and $p_{n+1} = \alpha$.

   (b) Normalize $\mathbf{p}$.

   (c) If any element in $\mathbf{p}$, for $j \in [1..n]$, is less than the corresponding element in $\mathbf{a}$, then rescale $\mathbf{p}$ as follows:

      i. Clamp the offending element p_j to a_j and rescale $\mathbf{p}$ over the unclamped elements.

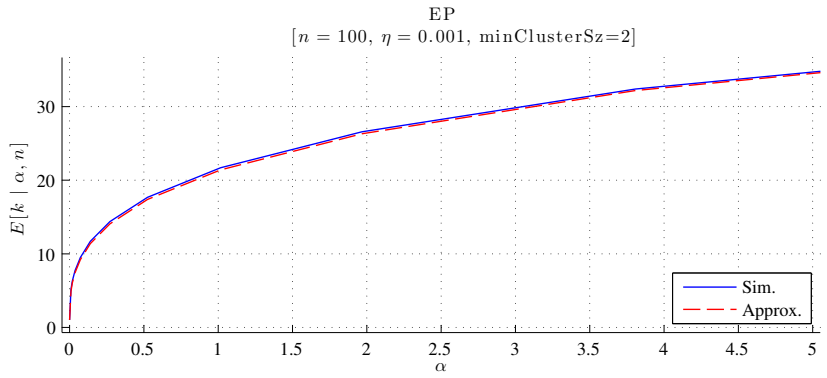
      ii. Repeat until either all elements in $\mathbf{p}$ are sufficiently sized or all have been clamped (and the remaining probability mass goes to p_{n+1}.

   (d) Update $\mathbf{a}$: $a_1 = a_1 + p_{n+1} - p_1$ and $a_j = a_j + p_{j-1} - p_j$ for $2 \le j \le n$.

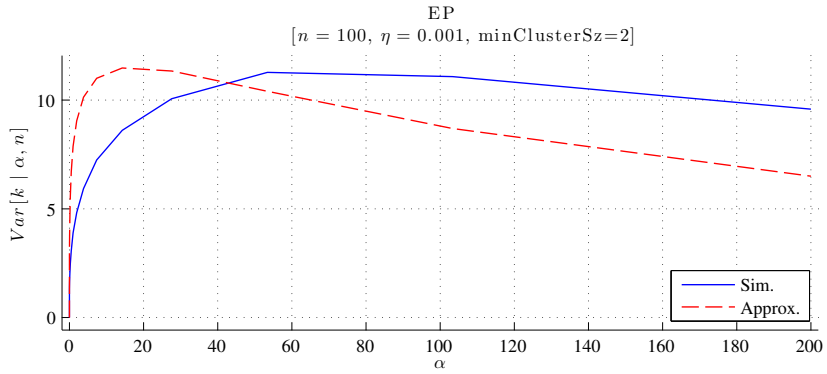
   (e) Update e: $e = e + p_{n+1}$.

   (f) Update v: $v = v + p_{n+1}\,(1 - p_{n+1})$.

Figures 3.19 and 3.20 show examples of how these estimates compare to the values obtained by simulation. For EP, the expected number of clusters is accurate to within a fraction of cluster. However, the variance fails to match the correct shape although it is broadly the correct magnitude. The mismatch in terms of standard deviation is on average 0.4 clusters with a maximum of 0.8 clusters. This is a consequence of the fact that the estimation procedure does not account for the covariance between the estimates of indicators. The estimation procedure could in principle be extended to estimate those terms. However, when we studied such extensions we found the improvement to be rather marginal due to both problems with numerical stability and the inherent inaccuracies in the simple estimation model. The current form represents a more attractive trade-off between cost and accuracy. For EP2, the expected number of clusters suffers a bit due to inaccuracies in the estimation model. Here too we looked into more accurate alternatives to the estimation model, but found them to yield too little gain for the increased cost. The maximum error, which occurs at $\alpha \approx 0.05$, is approximately 0.5 clusters and decreases as α increases. The variance is reasonably well-modeled.

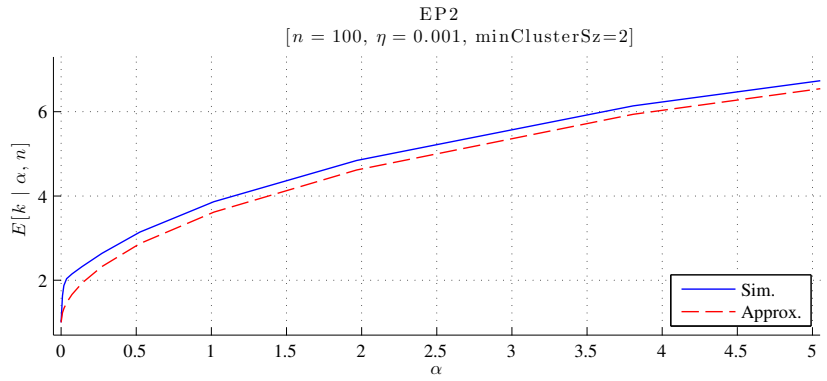


(a)

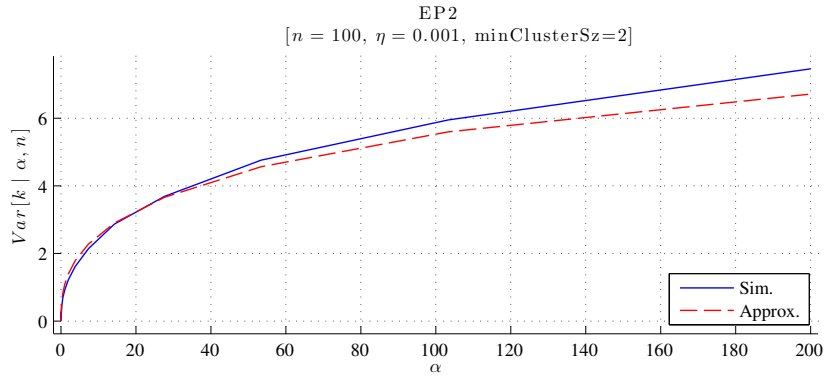


(b)

Figure 3.19: Accuracy of the estimated mean and variance of the number of clusters for the egalitarian process.



(a)



(b)

Figure 3.20: Accuracy of the estimated mean and variance of the number of clusters for the second form of the egalitarian process.

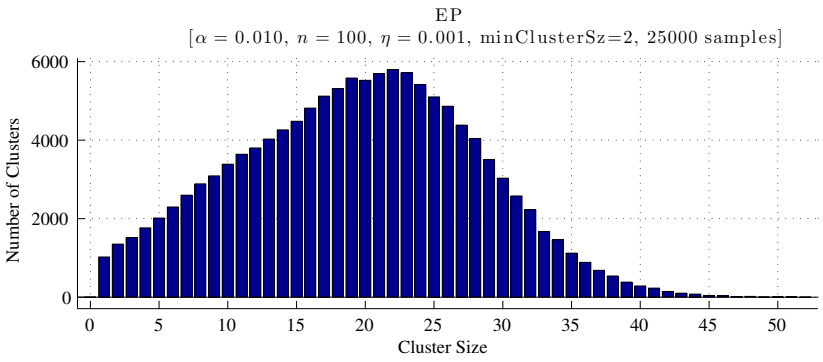


Figure 3.21: Distribution of clusters sizes produced by the egalitarian process.

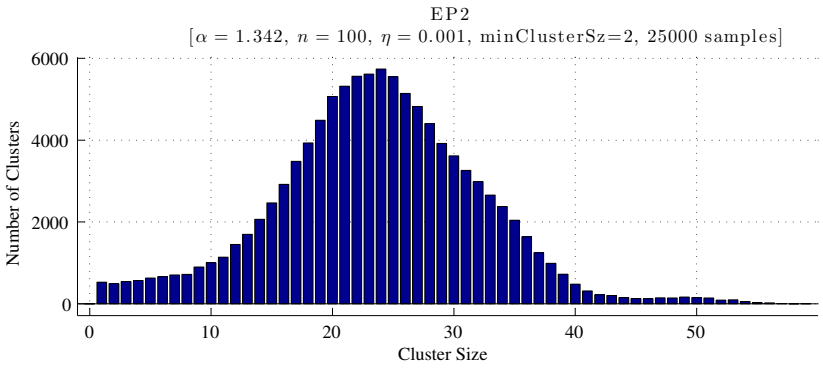


Figure 3.22: Distribution of clusters sizes produced by the second form of the egalitarian process.

Referring back to figure 3.1, we can see how the mean and variance of the number of clusters produced by the EP and EP2, as a function of α, compare to the other nonparametric processes.

As the prediction probability functions suggest, the EP and EP2 induce a distribution of clusters sizes that favors large clusters. Figures 3.21 and 3.22 show the distribution of clusters sizes for 25000 realizations of the respective processes with α set to its most likely value based on the hyperprior for the given η, *minClusterSz*, and n. For the EP, the preference is linear until the breakdown point. For the EP2, the preference is exponential.

3.4 Alternative Priors

3.4.1 Hierarchical Uniform Prior

Nonparametric models are not the only principled choice for the prior distribution of the target cluster labels. Casella et al. [26] developed one such alternative known as the hierarchical uniform prior (HUP). It was motivated by the desire to have an objective prior distribution over partitions that respects the structure of the clustering problem. That structure is described by the hierarchy: cluster class, configuration class, and partition.

The top of the hierarchy is the cluster class, which is simply the number of clusters. It is denoted by k. All partitions with k clusters belong to that cluster class. The next level is the configuration class, which we defined earlier. To reiterate, it is the class of all partitions with equivalent cluster size vectors. The configuration class is denoted by $\mathfrak{R}_{k;\mathbf{m}(n)}$, where the subscript indicates the corresponding cluster class and cluster sizes. Within each cluster class, k, there are one or more configuration classes, $\mathfrak{R}_{k;\mathbf{m}(n)}$, that contain all the partitions. Each of those configuration classes has a unique cluster size vector, $\mathbf{m}(n)$. Finally, there is the partition, which is given by the set of indicator variables, Z. Thus, each partition belongs to a particular configuration class that itself belongs to a particular cluster class.

To specify the prior, the joint distribution of the indicators is factored according to the hierarchy of the clustering problem to get:

$$p(Z) = p\Big(Z \mid \mathfrak{R}_{k;\mathbf{m}(n)}\Big)\, p\Big(\mathfrak{R}_{k;\mathbf{m}(n)} \mid k\Big)\, p(k \mid n)\,. \tag{3.21}$$

The first term is the probability of the partition given its configuration class. Since the difference between partitions within a given configuration class, ignoring the likelihood, is just the number of different ways to make to the assignments such that they have the correct cluster size vector, they are each assigned an equal probability. Thus, we have:

$$p\Big(Z \mid \mathfrak{R}_{k;\mathbf{m}(n)}\Big) = \binom{n}{m_1(n) \cdots m_k(n)}^{-1} R(m_1(n), \ldots, m_k(n))\,, \tag{3.22}$$

where $R(m_1(n), \ldots, m_k(n)) = \prod_{i=1}^{n} \Big[\sum_{j=1}^{p} I(m_j(n) = i)\Big]!$, which corrects the count for redundant assignments when $m_j(n) = m_l(n)$ for $j \neq l$.

The second term is the probability of the configuration class given its cluster class. These are assumed to be a priori equally likely. Thus, we have:

$$p\Big(\mathfrak{R}_{k;\mathbf{m}(n)} \mid k\Big) = b(n,\, k)^{-1}\,, \tag{3.23}$$

where $b(n,\, k)$ denotes the number of partitions of n with exactly k parts. It can be computed using the recurrence relation:

$$b(n,\, k) = b(n-1,\, k-1) + b(n-k,\, k)\,,$$

where $1 \leq k \leq n$. The base conditions are $b(n,\, 1) = 1$ and $b(n,\, n) = 1$.

Finally, the last term is the prior probability that n measurements will contain k clusters. Casella et al. suggest a couple of different ways to specify this prior based on the assumption that a small number of clusters is preferable. The result is the Poisson-Jefferys and Poisson-Intrinsic priors; see [26] for details. In either case, $p(k \mid n)$ is a distribution which decays exponentially from its peak at $k = 1$. The rate of decay, hence the thickness of the tails, is what differs between the two approaches. One could also use the formulation of the prior introduced in the last chapter (equation (2.25)) that parameterizes the decay in terms of the ratio of the most probable to the least probable.

By respecting the structure of the clustering problem, the hierarchical uniform prior avoids the often unforeseen consequences of a uniform prior that simply assigns the same probability to every partition. The uniform prior can lead to large differences between the probabilities of different configuration classes within the same cluster class, even when they differ by just a single assignment. These differences are based entirely on the number of partitions in each configuration class, rather than on some preferences regarding the different configurations. It turns out that a uniform probability with respect to the partitions actually implies some rather strong beliefs about the number of clusters. Figure 3.23 shows the distribution of the number of clusters implied by a uniform prior over partitions for $n = 20$. In terms of the number of clusters, the result is far from uniform or uninformative. Nor is it likely to represent any prior beliefs that lead us to select a "uniform" in the first place. In addition to these problems, the uniform prior was also found to be inconsistent with respect to model selection and hence should be avoided. The hierarchical uniform prior, on the other hand, is consistent on the space of models and allows us to explicitly state our prior beliefs about the number of clusters. In addition, compared to the nonparametric priors, it

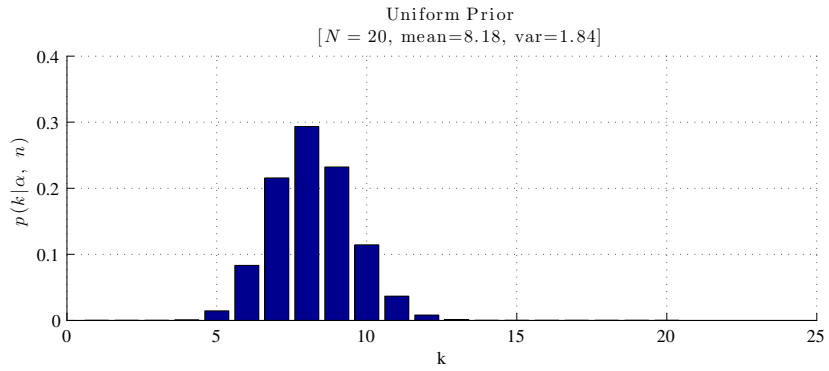


Figure 3.23: Distribution of the number of clusters implied by placing a uniform prior on the partitions.

has the advantage needing no hyperprior. One of the criticisms of the nonparametric priors is their sensitivity to the value of α.

3.4.2 Property-based Priors

Outside of the statistical literature, a common choice for the prior distribution over cluster labels is what we call a property-based prior. The term is meant to distinguish these priors, which are constructed based on notions of a partition's properties, from those that we looked at earlier. In contrast, nonparametric priors are constructed based notions of the clustering process that generates the partition while the hierarchical uniform prior is constructed based on notions of the distribution over partitions. Property-based priors are typically created by taking the product of various functions of the indicator statistics (e.g. number of clusters, cluster sizes, etc.) each designed to respond to some desired property of the partition. For example, the product of functions expressing a preference for the length of each track and minimal overlap (in the case of merged measurements) have been used [201, 99, 199]. They have also been created by combining functions, one per scan, that respond to the properties implied by detection and target appearance/disappearance models [135, 169]. Usually, the expected size of the false alarm cluster is also among the properties represented. However, we will ignore that here because our tracking model separates the prior over false alarms and the prior over the cluster labels.

Property-based priors are appealing because of their flexibility. Although it is not relevant to our needs, a prime example is the ability to combine specific beliefs about

the size of the false alarm cluster with general preferences for the target clusters. Note that existing statistical models of the data association problem that can handle false alarm measurements have used these priors exclusively; presumably for this reason. This flexibility comes from the fact that such priors simply ignore the combinatorial structure of the space of cluster labels. The problem with this approach is that while the preferences expressed by a property-based prior might achieve the desired effect when comparing two partitions, those preferences will not necessarily be reflected in the distribution over cluster labels. This happens when the strength of the responses used to define the prior are not calibrated to account for number of different label configurations that correspond to any given partition structure.

To illustrate the point, let us consider the distribution over partitions implied by several different property-based priors each intended to express the general belief that a smaller number of clusters is more likely than a larger number. The first prior is based on a preference for partitions with a small number of clusters, which represents a direct approach. It is defined using a function of k, the number of clusters, that decays exponentially. The rate of decay is set so that the smallest number of clusters is one million times more likely than the largest number of clusters. Figure 3.24a shows the resulting $\log p(Z)$ values when 20 clustering measurements. Note that the configuration classes are sorted so that the number of clusters grows along the x-axis and the tick marks indicate where each value of k begins. Despite the fact that the prior is locally consistent with our beliefs, the distribution over partitions clearly fails to reflect any preference for partitions with a small number of clusters. On the contrary, the most probable partition under this model contains 7 clusters with the likelihood generally increasing from 1 to 7 clusters and then generally decreasing as the number of clusters goes to 20. Instead, we might try a second prior based on a preference for partitions with large clusters. It is defined using an exponential function of the average cluster size. The rate of growth is set so that the largest average cluster size is one million times more likely than the smallest. The resulting distribution is shown in figure 3.24b. It too fails to represent our beliefs and is broadly similar to that from the first prior. Finally, let us try a third prior based on the same preference for partitions with large clusters. This time, however, it is defined by taking the product of an exponential function of each cluster's size. The rate of growth in each function is set so that the largest cluster size is one million times more likely than the smallest. Figure 3.24c

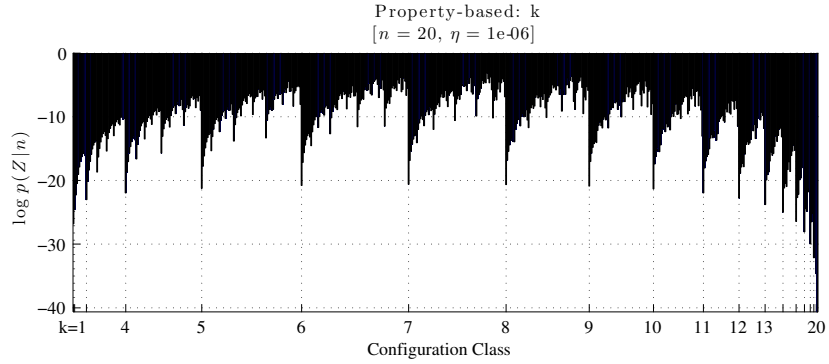
shows the resulting distribution and it is in fact consistent with our beliefs. However, the reason has less to do with the sized-based preference than with the structure of the prior itself. Since the probability of a given partition is determined by a product of probabilities, in which the number of terms grows with the number of clusters, small cluster sizes are implicitly preferred; at least locally. We could just as well have used a constant value for each of the product terms, independent of the cluster size. As long as that value is sufficiently small, the distribution over cluster labels will reflect our beliefs. The sized-based preference only amplifies the effect.

The flexibility of property-based priors comes at the cost of sensitivity to their parameter values. In the examples above, we could have strengthened the preference in the first two priors to produce a distributions over cluster labels that represented our belief that a smaller number of clusters is more likely than a larger number. The problem is that in practice we do not know how strong those preferences need to be. And as those first two examples clearly show, there can be more at stake than just the strength of our beliefs. Choose the wrong values for the parameters and we may very well be expressing beliefs quite contrary to our intention. Even seemingly reasonable parameter values might not work as expected because their impact on the implied distribution over cluster labels is difficult, if not impossible, to discern. As the third example illustrates, the structure of the prior plays a critical role in this relationship. Unfortunately, even a structure conducive to our beliefs can obscure the role of the preferences and the effect of the parameters. In general, specifying reasonable parameter values for property-based priors is extremely difficult and such priors should be used with caution.

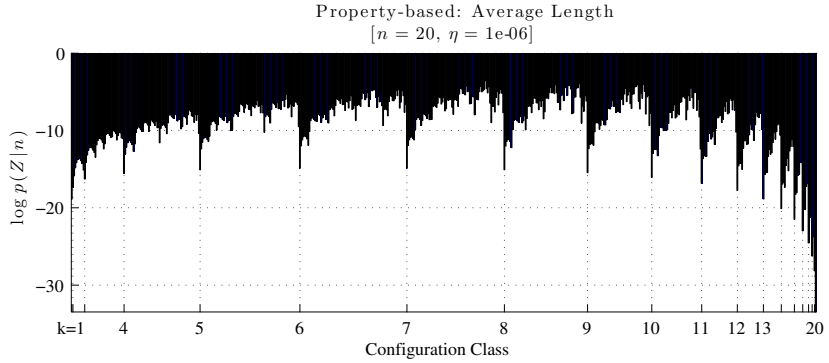
One exception is when the prior is based on the properties implied by detection and target appearance/disappearance models. In this case, the parameters have well-defined values and it is the structure of the prior that is the critical design decision. Here, we will consider the prior proposed by Oh et al. [135] in their seminal paper on Markov chain Monte Carlo data association, which is perhaps the most principled construction. Omitting the term pertaining to the false alarm rate, it is defined as:

$$p(Z) \propto \prod_{i=1}^{ns} p_z^{z_{s_i}} \, (1 - p_z)^{c_{s_i}} \, p_d^{d_{s_i}} \, (1 - p_d)^{g_{s_i}} \, \lambda_b^{a_{s_i}},$$

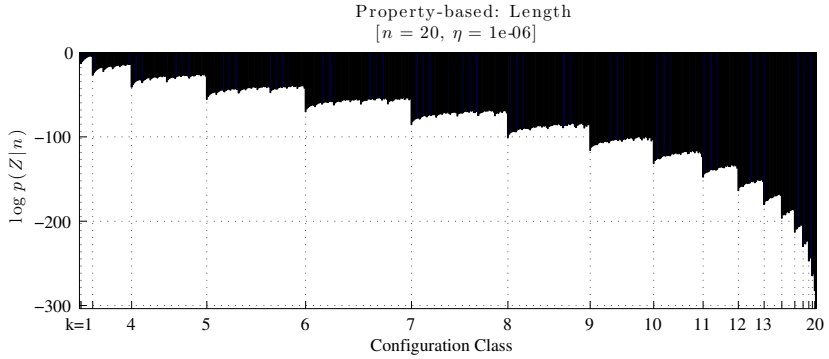
where p_z is the probability of a target disappearing during a given scan, p_d is the probability of detecting a target during a given scan, and λ_b is the birth rate of new targets per scan per unit volume (or area). The remaining variables are derived from



(a)



(b)



(c)

Figure 3.24: Distribution over partitions implied by several property-based priors.

Z where z_{s_i} denotes the number of previously detected targets that disappeared during scan i, c_{s_i} denotes the number of targets persisting from scan $i-1$, d_{s_i} denotes the number of targets detected during scan i, g_{s_i} denotes the number of targets not detected during scan i, and a_{s_i} denotes the number of new targets appearing during scan i.

Notice how the terms in the product are based on cumulative statistics of Z. Each term expresses the model-derived beliefs about the probability of the assignments in scan i given the previous assignments. In this way, its form is similar to the prediction rule that is used by the nonparametric priors to implicitly define a partition. Consequently, the resulting $p(Z)$ does not suffer from the same problems that we saw in the earlier examples of property-based priors. However, unlike those other priors, this one is not exchangeable. Despite that fact, the structure of the prior is still compelling. Unfortunately, the parameters that make it problematic to use in practice. While they certainly have well-defined values, those values are not all generally known. The probability of detection can reasonably be modeled, but the assumption that it is constant throughout the surveillance region is questionable. The birth rate and disappearance probability, on the other hand, are unknown is most applications. In fact, we began this investigation into suitable forms for the prior distribution over cluster labels with the goal of modeling our lack of knowledge in this regard. Contrary to that objective, λ_b and p_z represent very specific beliefs about both the number and lifetimes of the targets. The birth rate in this prior not only represents how many targets we expect to see after *ns* scans but how we expect their appearances to be distributed across the scans. We might consider just ignoring the matter of how the appearances are distributed and place a hyperior on λ_b, along the lines of what we did with α for the nonparametric priors. However, λ_b does not directly determine the expected number of targets implied by $p(Z)$ because that value is affected by the disappearance probability, p_z. So, not only must we explicitly state our beliefs about how long a target will persist, but those beliefs must be balanced against out beliefs about how many targets might be present. It is not clear how to do this in a way that adequately reflects our ignorance.

3.5 Scan Constraints

Up to this point, we have looked at various ways to define the distribution of the target cluster labels without considering the impact of the scan constraint. Recall from

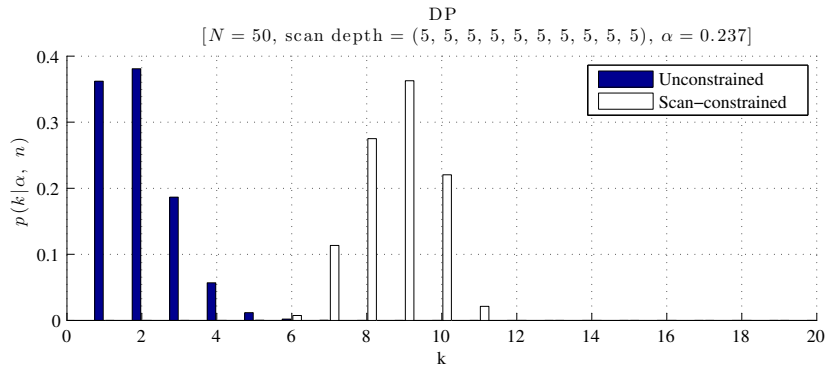


Figure 3.25: Impact of the scan constraint on the distribution of the number of clusters for the DP.

the previous chapter that the scan constraint dictates that no cluster can contain more than one measurement from any given scan. This implies that the a priori probability of any partition that violates the scan constraint is zero. Unfortunately, when the offending partitions are assigned zero probability, it changes the properties of the prior because it has effectively been truncated. For the nonparametric priors, this alters the relationship between α and the distribution of the number of clusters, on which the hyperprior depends. Although it is generally intractable to fully account for the effects of the scan constraint on these priors, we can generate reasonable approximations to the scan-constrained priors.

To illustrate the impact of truncating the support of $p(Z)$ based on the scan constraint, let us consider an example using the DP. Suppose that we collect five measurements in each of ten scans for a total of 50. Ignoring the scan constraint, we set α to its most likely value based on the hyperprior given $\eta = 0.001$ and $minClusterSz = 4$. Figure 3.25 shows the distribution of the number of clusters produced by simulating the DP both with and without the scan constraint. Ideally, the effect of the constraints would be to effectively shift the unconstrained distribution (the dark blue bars in the plot) to the right onto the truncated support of k, which is [5..50]. However, the constrained distribution (the white bars in the plot) is effectively shifted and skewed to the right, placing most of the probability on a larger number of clusters than our prior beliefs call for. To achieve the desired result, we need to be able to set α based how the DP behaves subject to the scan constraint. The impact and the remedy is similar for all the nonparametric priors that we have discussed.

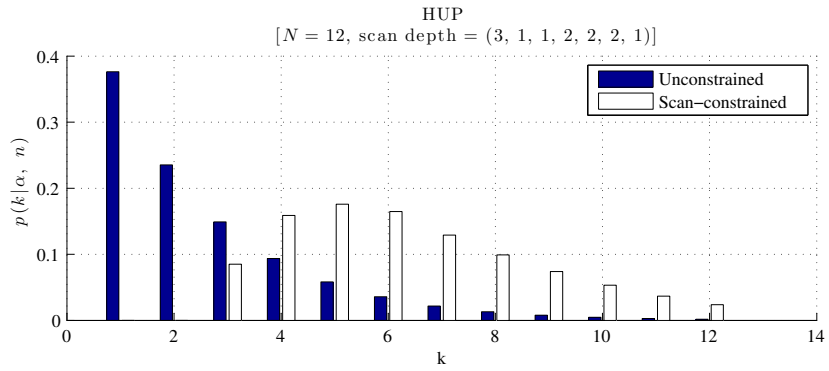


Figure 3.26: Impact of the scan constraint on the distribution of the number of clusters for the HUP.

Now let us consider how the hierarchical uniform prior is impacted by the scan constraint. This time we will use a smaller example because unlike the DP, which is a stochastic process and can be simulated, here we have enumerate all the partitions. Suppose we collect 12 measurements with the following scan depths: (3, 1, 1, 2, 2, 2, 1). Ignoring the scan constraint, there are 4213597 partitions. However, subject to the scan constraint only 1462981 have nonzero probability. Clearly, the hierarchical uniform prior will be affected because the partition counts in each level of the hierarchy will no longer be correct. Figure 3.26 shows the resulting distributions of the number of clusters. Note that the Poisson-Poisson-Intrinsic prior was used for the prior over k in the model. As before, we would ideally like to see the unconstrained distribution (the dark blue bars in the plot) shifted onto the truncated support of k, which is [3..12] in this case. Although the result is not completely unreasonable, it fails to match our stated prior beliefs. To achieve the desired result, we need to be able to modify the first and second terms in (3.21) to reflect the constrained partition counts.

Note that the property-based priors will not be considered in the context of the scan constraint. In most cases, the truncated support will further complicate the already thorny issue of how to set the parameters. The prior from Oh et al. [135], on the other hand, was built around the concept of scan-based data collection and is thus unaffected.

3.5.1 Scan-constrained Nonparametric Priors

For the nonparametric priors, we focus our attention on how the scan constraint alters the relationship between α and the distribution of the number of clusters. We assume that for an appropriately selected α, the EPPF can be used without modification. Thus, our objective is to estimate $p(k \mid \alpha, n)$ subject to the scan constraint.

We begin by recognizing that it is not necessary to work directly with the distribution of the number of clusters. Instead, we can simplify things by working with the moments: $E[k \mid \alpha, n]$ and $Var[k \mid \alpha, n]$. The distribution $p(k \mid \alpha, n)$ can then be estimated from these moments using either of the approximations introduced for the UP.

Looking first at $E[k \mid \alpha, n]$, as a function of α, the question is: how does the scan constraint change this behavior? Without the scan constraint, as $\alpha \to 0$ the expected number of clusters goes to one as the process tends to assign each indicator variable the initial cluster label. At the other extreme, as $\alpha \to \infty$ the expected number of clusters goes to n as the process tends to assign each indicator variable a new label. Subject to the scan constraint, it is no longer possible, in general, to assign each indicator variable the initial cluster label. Thus, the lower bound on the number of clusters is moved from one to the maximum scan depth. The upper bound remains the same.

This suggests shifting and scaling the expected number of clusters so that its limiting behavior is correct. Thus, we can write:

$$\begin{aligned} E[k \mid \alpha, n] &= (e-1)\,\kappa + K_{min} \\ Var[k \mid \alpha, n] &= v\,\kappa, \end{aligned}$$

where e and v are the unconstrained estimates of the mean and variance, respectively. The mean estimate is shifted by K_{min}, which is equal to the maximum scan depth. Both estimates are scaled by: $\kappa = \frac{n-K_{min}}{n-1}$.

Although this simple modification does not capture the full impact of the scan constraint, it gets us close. The expected number of clusters still does not grow fast enough for small values of α to match the behavior observed when simulating the scan-constrained process. The estimated variance, however, provides an adequate description of the scan-constrained behavior.

To improve the scan-constrained estimate of $E[k \mid \alpha, n]$, we can shift it by an additional amount $\hat{e} > 0$. This is combined with the following blending function to

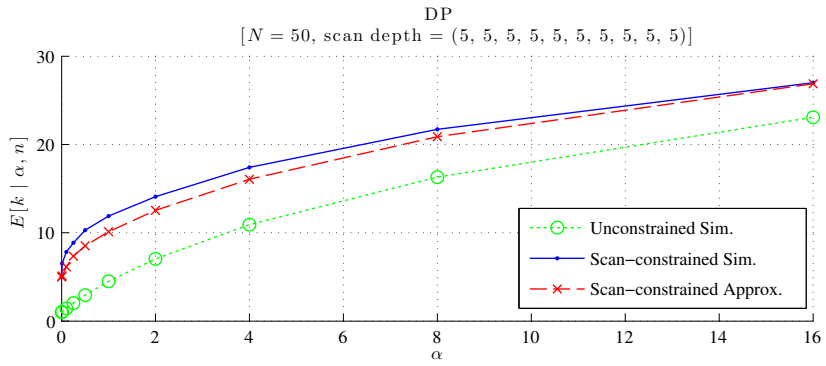


Figure 3.27: Impact of the scan constraint on the expected number of clusters, as a function of α, for the DP.

ensure the limiting behavior remains correct for $\alpha \to 0$:

$$\left(\frac{1}{1+\exp(-29\alpha)} - \frac{1}{2}\right) 2\hat{e}.$$

We accept that the limiting behavior for $\alpha \to \infty$ will now be biased since large values of α are not important for our application.

Empirically, we have determined that a reasonable way to estimate the additional shift, $\hat{e}$, is to set it to the value of α at which $E[k \mid \alpha, n] = K_{min}$ for the unconstrained process. Based on simulations for random scan depths with $1 \leq K_{min} \leq 10$, this appears to be a reliable shift value.

Figure 3.27 shows impact of the scan constraint on $E[k \mid \alpha, n]$ using our initial example with the DP. It also shows scan-constrained estimate computed using the aforementioned procedure. The growth rate can still be seen to lag a bit behind the simulated curve but the agreement has been greatly improved. If we now use the scan-constrained DP to set α to its most likely value based on the hyperprior, we get a much smaller value as expected. Figure 3.28 shows the resulting distribution of the number of clusters. Compared to 3.25, we can see that $p(k \mid \alpha, n)$ is close to the ideal but still overestimating by almost one cluster.

3.5.2 Scan-constrained Hierarchical Uniform Prior

As we stated earlier, the objective is to modify the first and second terms in (3.21) to reflect the constrained partition counts.

The first term, $p(Z \mid \mathfrak{R}_{k;\mathbf{m}(n)})$, is given by the product of the reciprocal of a

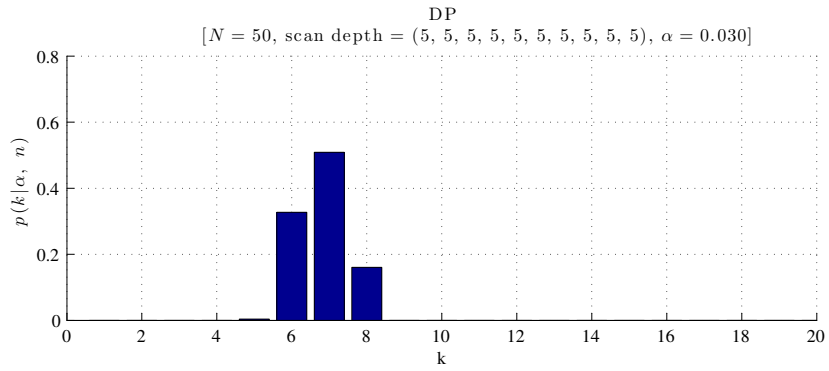


Figure 3.28: Distribution of the number of clusters using the scan-constrained DP.

multinomial coefficient and a correction factor, $R(\cdot)$. The correction factor remains the same when subject to the scan constraint, so we focus on the multinomial coefficient:

$$\binom{n}{m_1(n)\cdots m_k(n)}$$

which gives the number of ways to put n objects into k bins, with the number of objects in each bin given by $\mathbf{m}(n)$.

It is equal to the product:

$$\binom{n}{m_1(n)}\binom{n-m_1(n)}{m_2(n)}\binom{n-m_1(n)-m_2(n)}{m_3(n)}\cdots\binom{m_k(n)}{m_k(n)}, \tag{3.24}$$

where the terms are binomial coefficients that respectively give the number of ways to choose $m_1(n)$ from n, then choose $m_2(n)$ from the remaining, etc. Subject to the scan constraint, some fraction of these ways will be invalid. Unfortunately, there does not appear to be an efficient way to compute the correct numbers exactly. Once again, we must rely on an approximation.

As an analog to the binomial coefficient when the choices are subject to the scan constraint, we can use elementary symmetric polynomials. The elementary symmetric polynomial in p variables and of degree k is defined by:

$$e_k(X_1, \ldots, X_p) = \sum_{1\le j_1<j_2<\cdots<j_k\le p} X_{j_1}\cdots X_{j_k},$$

where $e_k(X_1, \ldots, X_p) = 0$ if $k > p$. Thus, when $(X_1, \ldots, X_p)$ is set to the scan depths, with p equal to the number of scans, it gives us the number of ways to choose k objects

subject to the scan constraint. The elementary symmetric polynomial can be computed using the algorithm in [90].

The problem is that this cannot be used as a direct replacement for the binomial coefficient terms in (3.24) because, although we know the number of remaining partitions after each choice, we do not know what the scan depths look like after each choice. What we can do instead is estimate the expected scan depths after each choice since the X_i need not be integer values.

After each choice, we update the expected scan depths as follows. First, we decide which depths will change. Any that are equal to number of remaining choices are decremented by one since these will have to be selected as part of every choice. The remaining number of selections for the given choice are spread over some subset of the unselected depths. This subset is determined by first selecting one of each unique depth. If there are not enough unique depths to cover the remaining selections, then we take two of each unique depth (when possible) or three of each unique depth, etc. We stop when there are at least as many depths selected as the number of remaining selections. Each depth in the subset is decremented by the appropriate fraction of remaining selections. Although clearly not exact, this procedure does a reasonable job in practice.

The second term, $p(\mathfrak{R}_{k;\mathbf{m}(n)} \mid k)$, is given by the reciprocal of the number of partitions of n with exactly k parts. Subject to the scan constraint, some of these partitions will be invalid. Thus, the scan constraint effective restricts the partitions. The primary restriction is that no part can be larger than the number of scans.

The number of partitions of n with exactly k parts and each part $\leq m$ is given by: $b(n,\, k,\, m) - b(n,\, k-1,\, m)$ [1]. It can be computed using the recurrence relation:

$$b(n,\, k,\, m) = b(n-k,\, k,\, m-1) + b(n,\, k-1,\, m)\,,$$

where $1 \leq k \leq n$ and $1 \leq m \leq n$. The base conditions are:

$$\begin{aligned} b(n,\, k,\, m) &= 0 \quad \text{if } n < 0 \text{ or } b = 0 \text{ or } kb < n \\ b(n,\, k,\, m) &= 1 \quad \text{if } n = 0 \text{ or } kb = n. \end{aligned}$$

This alone provides a very good estimate of scan-constrained number of partitions. However, it will overestimate the count in some cases because the maximum part size is not the only restriction implied by the scan constraint. Unfortunately, the only way to account for the full impact of the scan constraint would be to enumerate

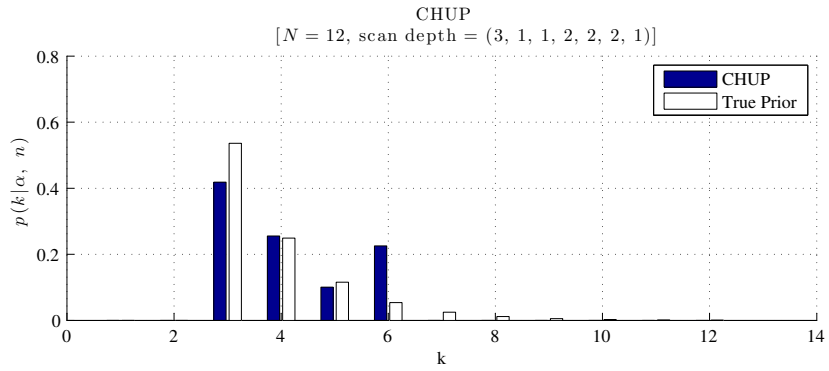


Figure 3.29: Distribution of the number of clusters using the scan-constrained hierarchical uniform prior.

the valid partitions. By generating them in lexical order and using various heuristics to eliminate invalid partitions before generating them completely, this is a tractable approach for realistic data sets but very computationally expensive. We do not believe that the benefits outweigh the costs in this regard.

Returning to our initial example with the hierarchical uniform prior, figure 3.29 shows the scan-constrained distribution of the number of clusters computed using the aforementioned procedure. It can be seen to overestimate the probability of $k = 6$ but otherwise faithfully represents our prior beliefs.

Chapter 4

Application to Surveillance of Dismounts

4.1 Introduction

The current research was motivated in part by applications involving the wide-area surveillance of human targets using radar. And this particular combination of sensor and target presents some challenges.

First, the targets are observed simply as points in space and time. There is no appearance information on which to rely when associating observations. Thus, targets must be discriminated based on their dynamics. Second, the targets are observed on a coarser time scale than typical radar tracking applications. This is due primarily to their inherent difficulty to detect at long range using radar. Since people have relatively small radar cross-sections, they reflect only a small amount of energy back to the sensor making it hard to maintain an acceptable probability of detection at a manageable false alarm rate. In addition to a weak return, their relatively low speed makes human targets especially difficult to spot against the ground clutter. To compensate, some systems slow the scan rate in order to put more energy on the target. Others integrate information from multiple scans before reporting a detection. Spatially varying probabilities of detection that are less than one, including the extreme case of target occlusion, also contribute to the sparsity of the measurements. Finally, the targets are highly maneuverable. Unlike aircraft, ships, or even vehicles, there are relatively few constraints on the ability of people to change their speed or direction between observations.

Individually, radar sensors and human targets have been the subjects of a considerable amount of research. However, there is currently very little that addresses these specific requirements. Existing models for people tracking tend to rely heavily on appearance information, with the target dynamics playing a secondary role. Most of this work has been for short range applications with frequent measurements. Until relatively recently, radar surveillance of ground targets was largely limited to vehicles and people tracking at long range was more or less the exclusive domain of EO/IR sensors. However, this is rapidly changing. Technological advances, especially with respect to the processing power available to the detector, have made radars with dismount detection capability increasing common.

Here, we will start to address some of the issues related to the wide-area surveillance of human targets using radar. We begin by studying the dynamics of human walkers observed at this scale. This information will then be used to guide the

construction of dynamic models for use with the tracking algorithm. We will consider several different approaches to specifying such models. In the next chapter, we will characterize the performance of these models, including quantify the potential gains from state estimation assuming no ambiguity in the data association.

4.2 Human Walkers

4.2.1 GPS Data Collection

In order to study the dynamics of the target, we used a differential global positioning system (DGPS) to make precise measurements of the position, speed, and heading of a person walking in representative terrain. The subject wore a backpack containing the GPS receiver, data recorder, and a pole-mounted external antenna. Three different terrain types were used for comparison and two independent data collections were made for each terrain type.

Regarding the accuracy of the position measurements, a previous calibration of the unit determined that the base accuracy was 0.32 m. We modeled the uncertainty by assuming that the error, when projected onto the UTM grid, is symmetric and follows a Gaussian distribution in which 95% of the values fall within the stated accuracy. For each measurement, the base accuracy was scaled by the horizontal dilution of precision (HDOP) to reflect errors introduced by the geometry of the satellites. The result was then divided by two to determine the standard deviation of the error. The accuracy of the speed measurements was also modeled as Gaussian. The manufacturer specified that the speed was accurate to 0.0447 m/s, which was then divided by two to determine the standard deviation. Note that this agrees almost exactly with the standard deviation for walking speeds estimated in previous DGPS studies of human locomotion [173]. The accuracy of both the position and speed measurements is deemed sufficient to serve as ground-truth for this application. Heading measurements, on the other hand, will only be used as a rough guide since no information regarding their accuracy was available.

Data was collected at three sites selected to represent flat, rolling, and mountain terrain. In addition to the terrain itself, each particular site was chosen to provide ideal conditions for the data collection: an unobstructed view of the sky and proximity to a differential beacon. The sites and collection details are summarized below.

Flat Terrain (T1)

This site consisted of a mix of flat paved and crushed gravel trails. There was no appreciable elevation change. The first collection lasted approximately 45 minutes with a mean of 6.8 visible satellites and a mean HDOP of 1.34. The second collection lasted approximately one hour with a mean of 7.1 visible satellites and a mean HDOP of 1.22.

Rolling Terrain (T2)

This site consisted of a mix of dirt and crushed gravel trails. Some were flat but most wound through rolling hills. The total elevation change was on the order of 30 feet. The first collection lasted approximately one hour with a mean of 7.9 visible satellites and a mean HDOP of 1.4. The second collection lasted approximately 90 minutes with a mean of 7.7 visible satellites and a mean HDOP of 1.07.

Mountain Terrain (T3)

This site consisted of dirt single track trails along mountain top ridges. The total elevation change was on the order of 300 feet. The first collection lasted approximately two hours with a mean of 7.8 visible satellites and a mean HDOP of 1.9. The second collection lasted approximately 90 minutes with a mean of 7.6 visible satellites and a mean HDOP of 1.09.

4.2.2 Analysis

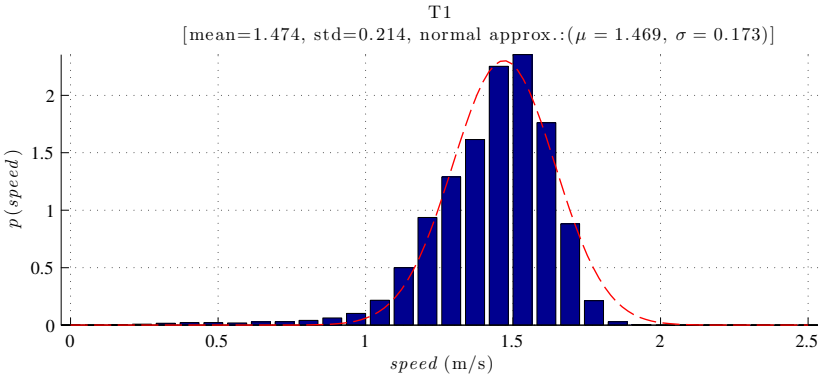
Here, we analyze the GPS data to look at the statistical properties of the speed and heading of a typical target in the given terrain. We will use the position measurements later, when we evaluate the performance of the dynamic models.

The probability density functions were estimated using kernel density estimation (KDE) with a Gaussian kernel and automated bandwidth selection. The autocorrelation functions (ACFs) were estimated using Welch's method with a 5 min window and 50% overlap. These calculations were all performed on measurements of steady-state walking, which we have defined as sequences of at least two minutes where the recorded speed is $> 0.25\,\mathrm{m}/$. Thus, we have filtered out any stops to focus specifically on moving targets.

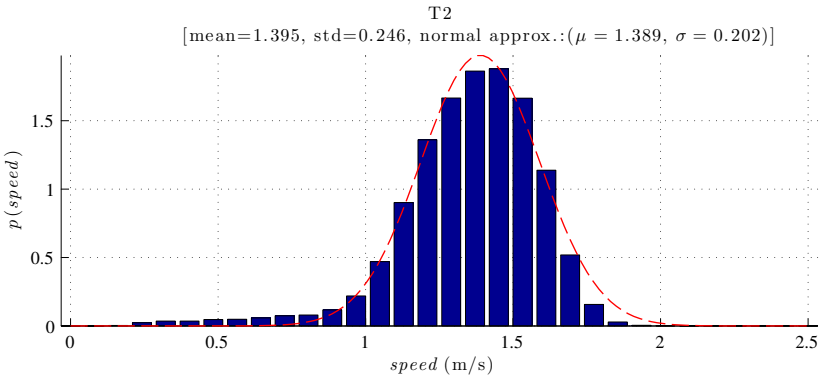
Figure 4.1 shows the distributions of target walking speeds for each of the three terrain types. For each terrain, data from both collections was combined to estimate the distributions. The closest, in the mean squared error sense, Gaussian approximation is also shown for reference. As we would expect, the distributions are all unimodal with the upper limit of their support at approximately 2 m/s, a point above which it is difficult for a person to maintain a walking gait and it becomes more efficient to run. The average expected speed across all terrains is approximately 1.4 m/s. For the flat terrain, the mode and expected speed are the highest. The distribution is skewed to the right so more of its mass is concentrated on the higher speeds. As the terrain becomes more rugged, the mode and expected speed drop. The skewness declines as well, but the variance increases. In mountain terrain, the tails of the distribution on both sides of the mean have balanced out and the distribution is approximately normal. Figure 4.2 shows the speed distributions for each terrain together on a single axis.

In practice, the continuous state of the target is sampled at discrete times. Figure 4.3 shows the distributions of speed changes for some typical sample periods, for each of the three terrain types. As we would expect, the distributions are essentially zero-mean. The magnitude of the changes are on the order of 0.25 m/s with the specific variation being a function of both the terrain and sample period. The terrain type is the strongest determinant of the magnitude of the speed changes with the largest changes occurring in the mountain terrain. For a given terrain, we observe a slight increase in the magnitude of the speed changes as the sample period increases. This suggests that the impact from autocorrelation decreases over time. If we assume that the speed is a wide-sense stationary process, with mean and variance equal to the values reported in figure 4.1, the speed change magnitude is consistent with its theoretical value given a relatively small autocorrelation (< 0.5) that decreases over time.

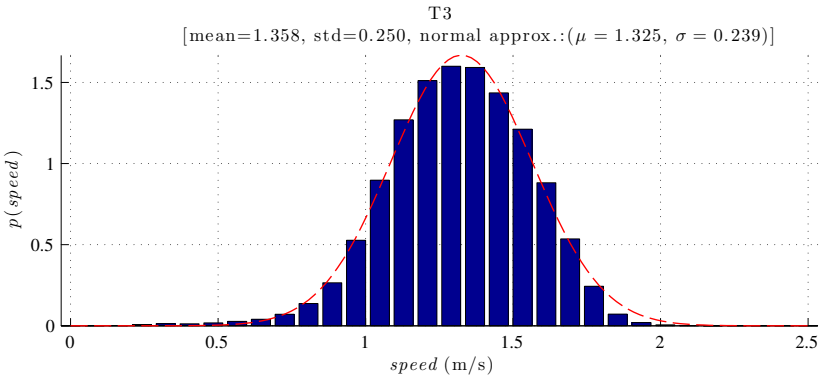
Looking at the autocorrelation of the walking speed, figure 4.4 shows that the shape and degree are consistent with what the speed change distributions suggested under the assumption of wide-sense stationarity. After about 30 s the speed is effectively uncorrelated for all terrains. The correlation time is longest in flat terrain as we would expect. In mountain terrain, the correlation time is shortest with the speed is largely uncorrelated after about 10 s. Consequently, we should not expect to be able to make good linear predictions about a target's speed in our application because of the coarse time scales involved. With the 30 s and 60 s sample periods, even assuming that the



(a) Flat terrain (T1).



(b) Rolling terrain (T2).



(c) Mountain terrain (T3).

Figure 4.1: Distribution of steady-state walking speeds.

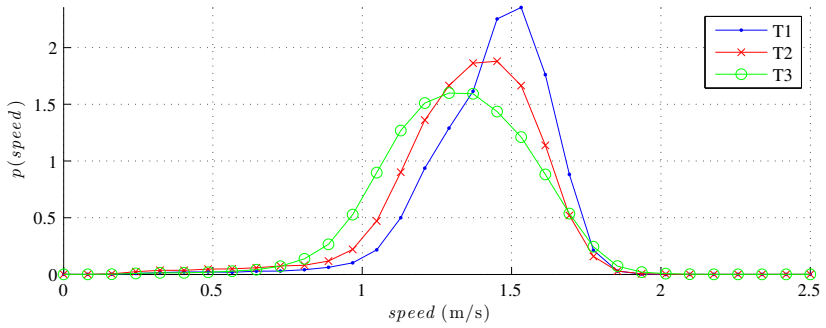
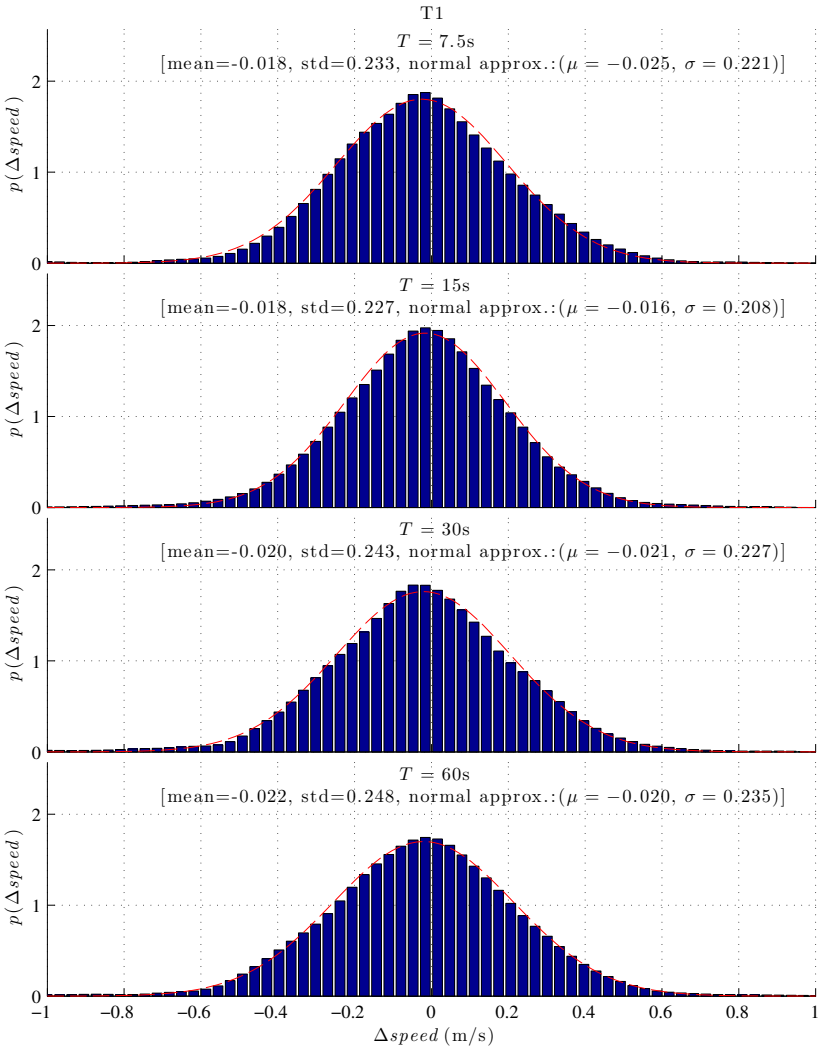


Figure 4.2: Comparison of steady-state walking speed distributions in flat (T1), rolling (T2), and mountain terrain (T3).

probability of detection is unity, we're already beyond the point where the current target speed has a significant linear relationship to the previous target speed. For the shorter sample periods, it will only take a few missed detections before the previous target speed will tell us little about its current speed. At this point, we may be better off making our prediction based on what know about the expected speed of a target given the terrain.

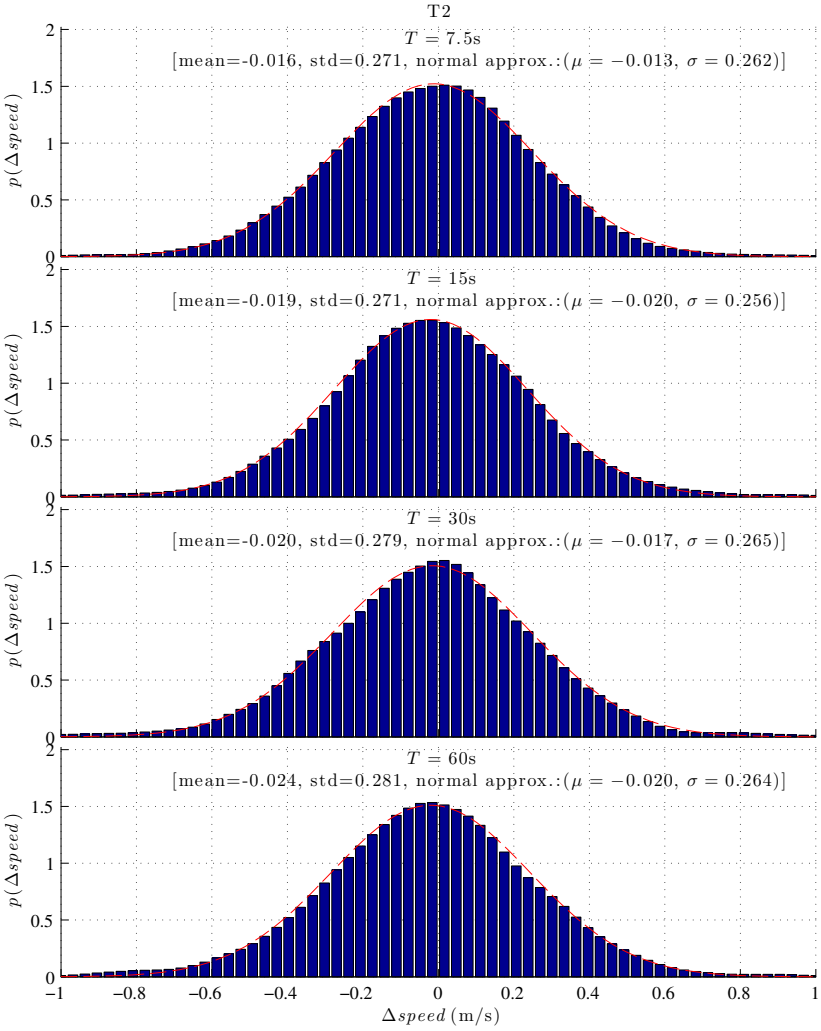
Turning now to the heading, we begin by noting that these measurements cannot be analyzed directly as was done with the speed. Unlike the time series of speed measurements, the heading measurements follow a clear trend that is dictated by the route. Thus, the distribution of target headings would tell us more about the route than the characteristics of the target. However, we can remove this trend and analyze the residual series to get some insight into the statistical properties of the heading independent of the route. To de-trend the series we subtract an estimate of the mean heading computed using a center-weighted moving average with a window of 30 s. This yields what we will call the zero-mean target headings.

Figure 4.5 shows the distributions of the zero-mean target headings for each of the three terrain types. The amount of variation is comparable across the different terrains with an average standard deviation of approximately 13.5°. Figure 4.6 illustrates this clearly by showing all three distributions together on a single axis. The most prominent difference between the distributions is how the tails get heavier as the terrain gets less flat. For the mountain terrain, the distribution is closer to a t-distribution than to a Gaussian. This does seem reasonable in light of the fact that rugged terrain places more restrictions on the directions that a target can move in. The single track at the mountain test site indeed favors consistent headings. Although these tendencies do



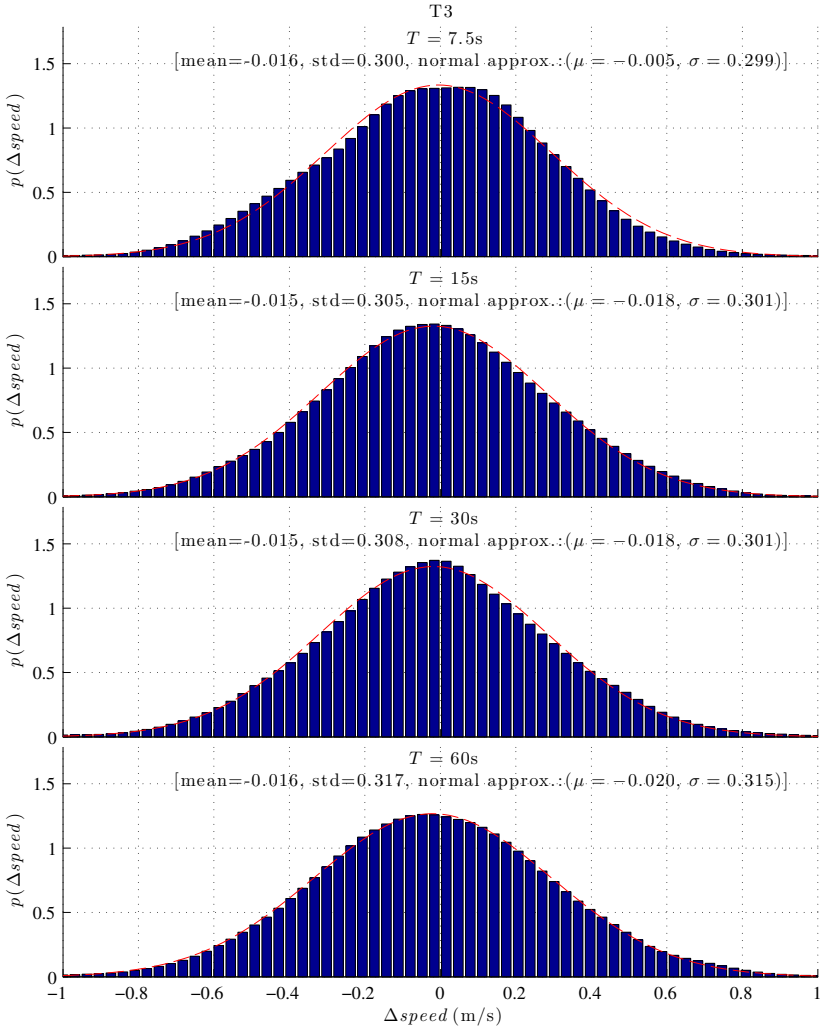
(a) Flat terrain (T1).

Figure 4.3: Distribution of speed changes for various sample periods.



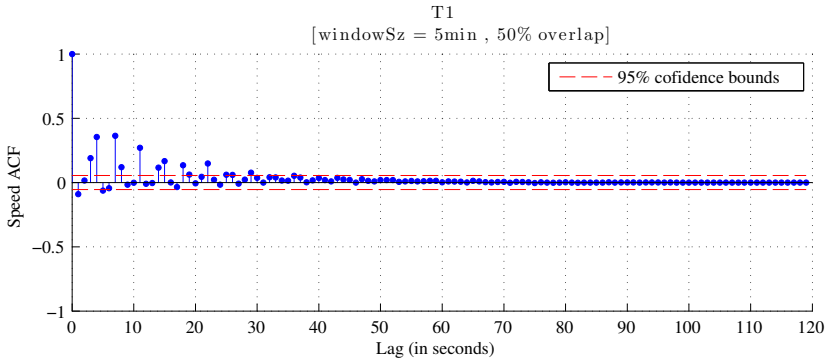
(b) Rolling terrain (T2).

Continuation of Figure 4.3: Distribution of speed changes for various sample periods.

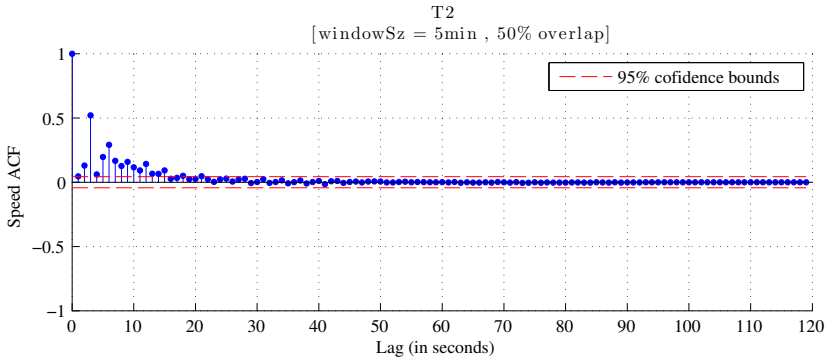


(c) Mountain terrain (T3).

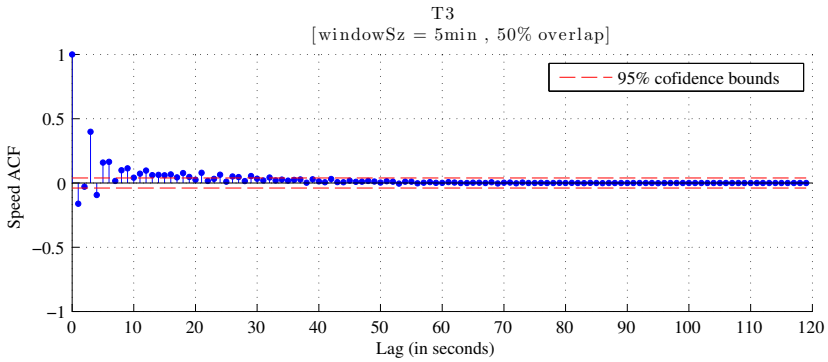
Continuation of Figure 4.3: Distribution of speed changes for various sample periods.



(a) Flat terrain (T1).



(b) Rolling terrain (T2).



(c) Mountain terrain (T3).

Figure 4.4: Autocorrelation of the steady-state walking.

make sense, we do not want to over analyze them because we lack an uncertainty model for the heading measurements and are relying on an estimate of the mean heading.

The distribution of zero-mean heading changes for some typical sample periods, for each of the three terrain types, are shown in figure 4.7. They reveal a small negative bias which appears to be a function route and introduced by the de-trending. The magnitude of the changes are on the order of 14.5° with the specific variation again a function of both the terrain and sample period. As was the case for the speed, the terrain type is the strongest determinant of the magnitude of the heading changes with the largest changes occurring in the mountain terrain. This result is somewhat surprising given that the mountain terrain places more constraints on the movement of the target. However, it is presumably an artifact of imperfect de-trending as the routes in mountain terrain will typically contain more twists and turns. For a given terrain, the magnitude of the changes do not follow a monotonic trend as the sample period increases. This suggests that there is some periodicity in the autocorrelation. However, assuming that the zero-mean heading is a wide-sense stationary process (which implies that the headings are trend stationary), the heading change magnitude is consistent with its theoretical value given a relatively small autocorrelation.

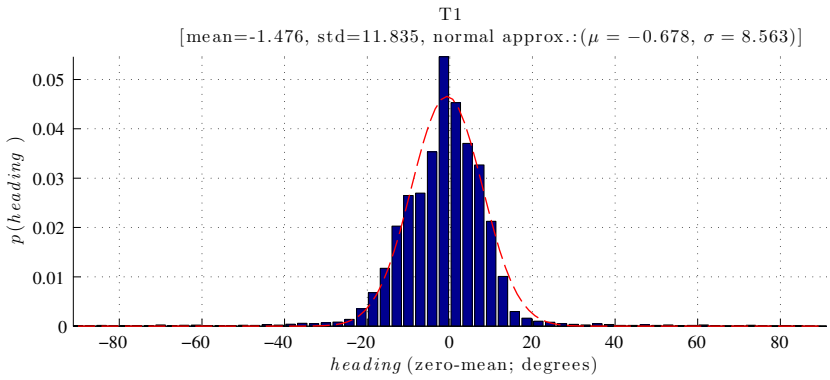
In figure 4.8, we can seen that the autocorrelation of the zero-mean headings is consistent with what the heading change distributions suggested both in terms of the degree and periodicity. The periodicity appears to be another artifact of the de-trending process. In any event, after about 40 s the heading is effectively uncorrelated for all terrains. As was the case for the speed, the correlation time is longest in flat terrain and shortest in mountain terrain, as expected. In mountain terrain, the heading is largely uncorrelated after about 15 s.

4.3 Dynamic Models

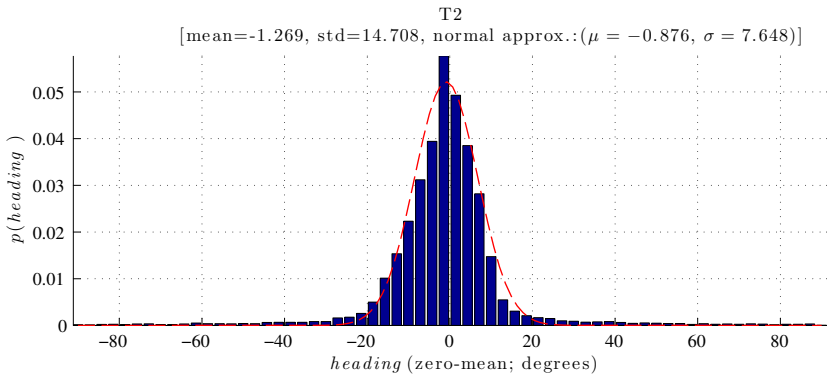
4.3.1 Preliminaries

Here, we will consider the problem of developing kinematic models for human motion observed at relatively large time-scales (on the order of 5 seconds to several minutes). In particular, our focus shall be restricted to human walkers, including the possibility of stoppage.

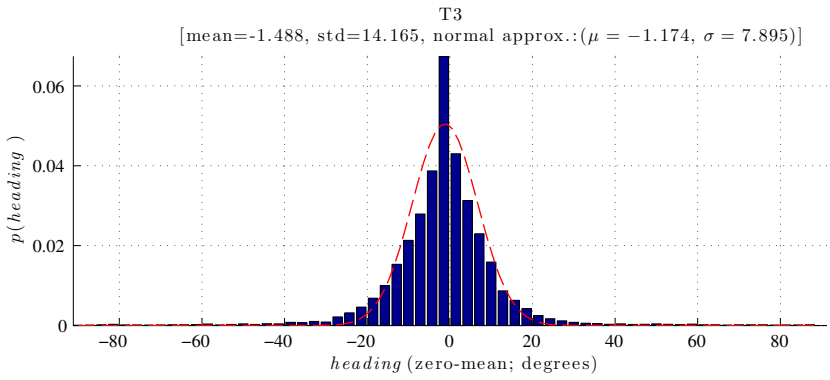
We begin with the following assumptions:



(a) Flat terrain (T1).



(b) Rolling terrain (T2).



(c) Mountain terrain (T3).

Figure 4.5: Distribution of zero-mean headings for steady-state walking.

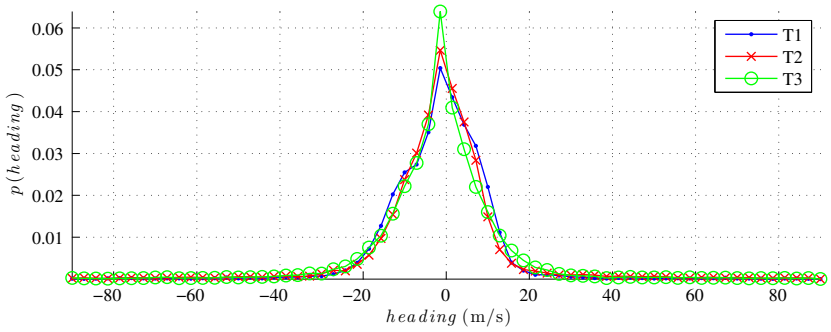


Figure 4.6: Comparison of zero-mean headings for steady-state walking in flat (T1), rolling (T2), and mountain terrain (T3).

- Human walkers are highly maneuverable.
- Restrictions to maneuverability are primarily due to environmental factors (e.g. terrain) or the target's goal/intent.
- The sampling period T is much larger than the maneuver time τ_m.
  - Between consecutive measurements, targets can start, stop, change to any heading or hit any speed within their range.
- Target state consists of 2-D position and velocity (which may be represented in terms of speed and heading).

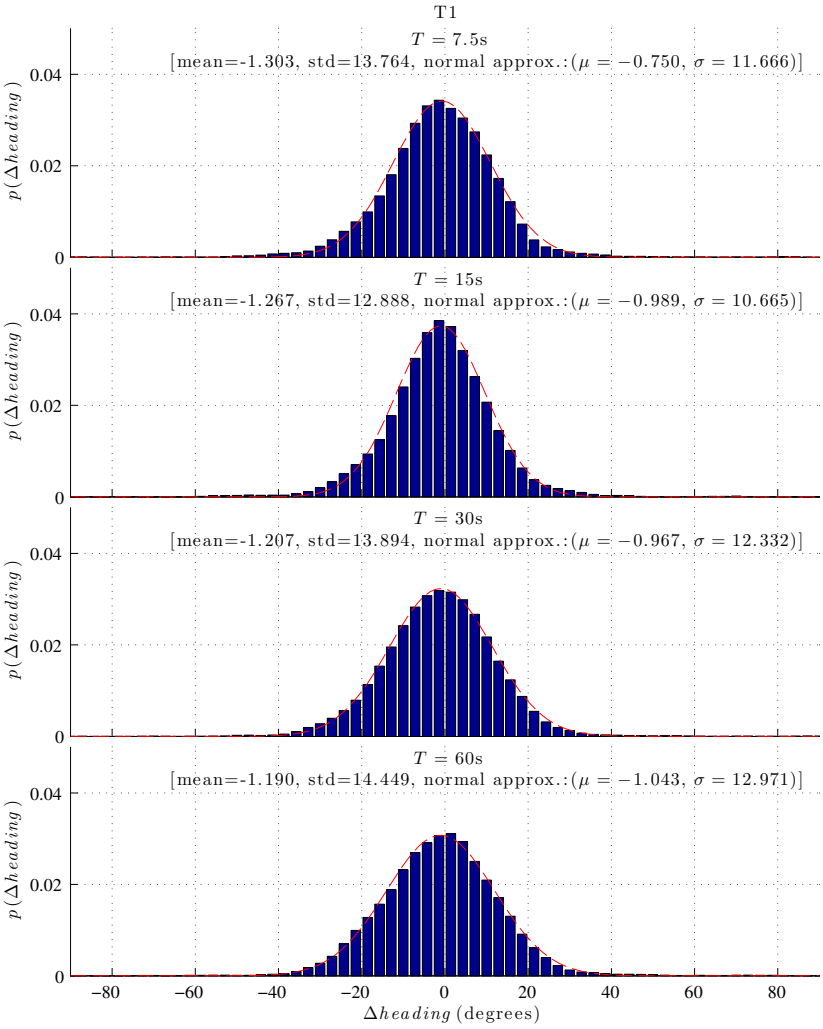
The models themselves must be linear-Gaussian (or linearized-Gaussian) models so that the state can be integrated out analytically as required for associative tracking.

Note that square brackets will be used to denote discrete-time variables to avoid confusion.

4.3.2 Target Speed

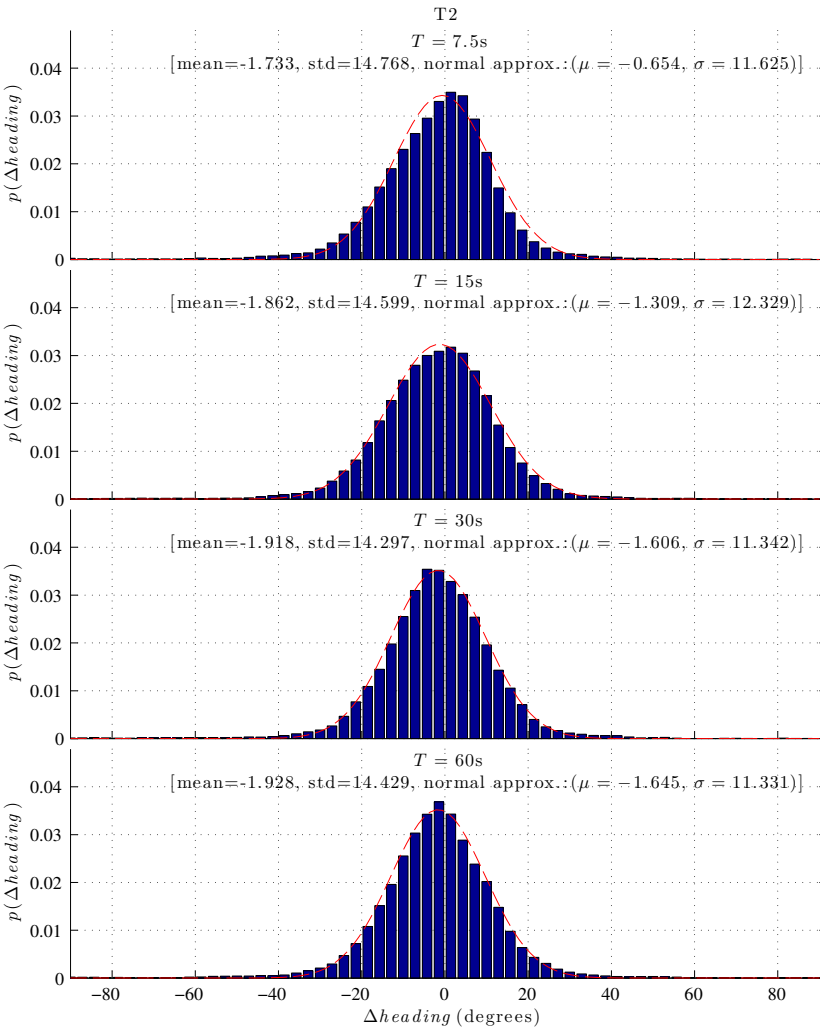
4.3.2.1 Models

Kinematic models for target tracking are typically derived from the equations of motion by representing a certain derivative of the position using a stochastic process. When the state consists of position and velocity, as we have assumed, the usual approach is to model the acceleration of the target with some type of noise process. This procedure



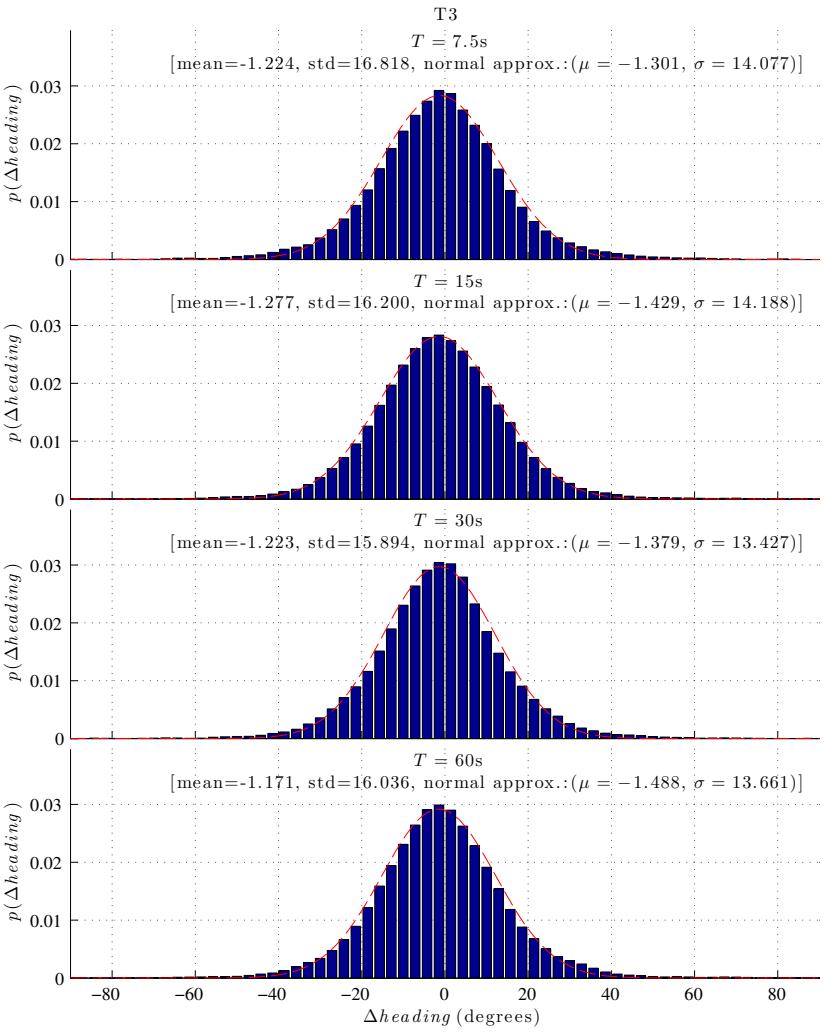
(a) Flat terrain (T1).

Figure 4.7: Distribution of zero-mean heading changes for various sample periods.



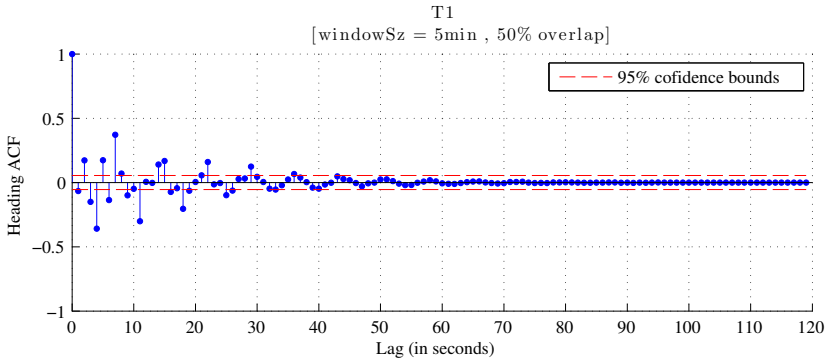
(b) Rolling terrain (T2).

Continuation of Figure 4.7: Distribution of zero-mean heading changes for various sample periods.

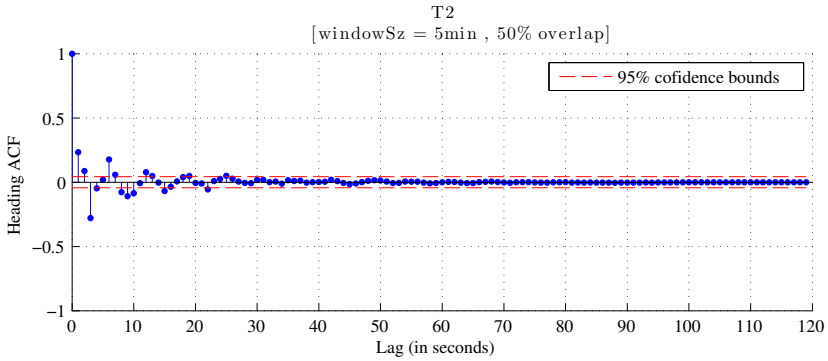


(c) Mountain terrain (T3).

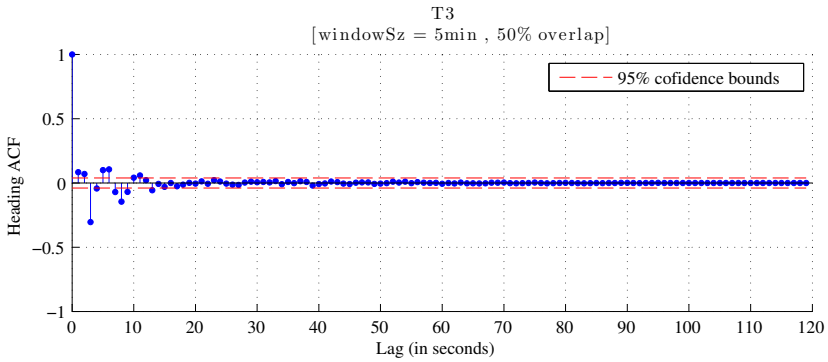
Continuation of Figure 4.7: Distribution of zero-mean heading changes for various sample periods.



(a) Flat terrain (T1).



(b) Rolling terrain (T2).



(c) Mountain terrain (T3).

Figure 4.8: Autocorrelation of the zero-mean heading for steady-state walking.

yields a second-order model of the continuous-time dynamics. Taking the integral of the model over the sample period then gives us the necessary discrete-time dynamics.

The integral of the acceleration noise process is what defines the velocity model, which is the primary means by which we characterize the motion of the target. Specifically, our focus will be on the magnitude of the velocity and how to represent it in a way that conforms to what was learned from the GPS study. Thus, we will refer to this as a "speed" model with the understanding that it technically describes the target's velocity since the sign imparts the direction of travel. This terminology is intended to emphasize the fact that these models are based on considerations of the magnitude of the velocity. The important question of the direction of travel will be taken up later.

Here, we will consider three different approaches to modeling the time-varying target speed, particularly with respect to the form of autocorrelation. Each will be presented as a simple linear dynamical system that describes the 1-D motion of a target using a state-space consisting of position, x, and velocity, v. The reduction to one dimension is intended to simplify the presentation. The system can be extended in a straightforward manner, by replicating the model for example, to describe 2-D motion.

The first system models target speed as a standard (i.e. no drift) Wiener process. The second system models the speed using an Ornstein-Uhlenbeck (OU) process with a known constant mean. Finally, the third system models the speed as Gaussian white noise process with a known constant mean.

Wiener Process Speed

This model is based on the assumption that a target's speed is constant except for stochastic variability in the form of acceleration noise. It is the well-known continuous white noise acceleration (CWNA) model [6], which can be regarded as the standard approach when velocity measurements are not available. We consider it here as a baseline due to both its simplicity and ubiquity.

The continuous-time system is given by:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t), \tag{4.1}$$

where $w(t)$ is a zero-mean Gaussian white noise process with variance σ_w^2.

The equivalent discrete-time system with sample period T is given by:

$$\begin{bmatrix} x[t+T] \\ v[t+T] \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x[t] \\ v[t] \end{bmatrix} + w[t], \tag{4.2}$$

where $w[t] \sim N(0, Q_w[t])$ with

$$Q_w[t] = \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \sigma_w^2. \tag{4.3}$$

From the continuous-time description of the system we can see that the speed is modeled as the integral of a Gaussian white noise, or equivalently, a Wiener process.

Since the Wiener process is not stationary, even the the wide sense, the speed time series does not have a stationary autocorrelation function.

Ornstein-Uhlenbeck Process Speed

Although the OU process first came to prominence as a model of the velocity of a Brownian particle, it is attractive as a speed model because the stationary autocorrelation function of an OU process decays exponentially with a rate that can be set directly using one of the parameters. Also of interest is the fact that the OU process is mean-reverting. These two properties are related and the OU process relaxes exponentially to its mean at the same rate as the autocorrelation decays.

We expect that in certain circumstances, the ability to explicitly model correlation in the target speed, even in this one simple form, will be beneficial. Figure 4.4 suggests that this may be useful for the dismounted targets in our application, although how useful will depend on sparsity of the measurements. For applications that include vehicular targets, the benefit is likely to be significant.

This model is based on the assumption that a target's speed, although stochastic, tends to drift over time toward some known mean value which might be conditional on the target's intent or by environmental factors such as terrain. We can think of the mean value as the target's preferred speed.

The continuous-time system is given by:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{\tau} \end{bmatrix} \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 1 - \exp\left(\frac{t}{\tau}\right) \\ \frac{1}{\tau} \exp\left(\frac{t}{\tau}\right) \end{bmatrix} u_v(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t), \tag{4.4}$$

where τ is the relaxation time constant, $u_v(t)$ is a control input (assumed to be constant throughout the sample period) that sets the mean speed value, and $w(t)$ is a Gaussian white noise process with variance σ_{ou}^2. The control gain is defined so that the impact of the constant input when integrated over $[t,\, t+T]$ is to set the mean speed.

The equivalent discrete-time system with sample period T is given by:

$$\begin{bmatrix} x[t+T] \\ v[t+T] \end{bmatrix} = \begin{bmatrix} 1 & \tau\left(1-\exp\left(-\frac{T}{\tau}\right)\right) \\ 0 & \exp\left(-\frac{T}{\tau}\right) \end{bmatrix} \begin{bmatrix} x[t] \\ v[t] \end{bmatrix} + \begin{bmatrix} T \\ 1 \end{bmatrix} u_v[t] + w[t] \tag{4.5}$$

where $w[t] \sim N\left(0,\, Q_{ou}[t]\right)$ with

$$Q_{ou}[t] \;=\; \begin{bmatrix} q11 & q12 \\ q12 & q22 \end{bmatrix} \sigma_{ou}^2 \tag{4.6}$$

and

$$q11 \;=\; \frac{\tau^3}{2}\left[-3-\exp\left(-2\frac{T}{\tau}\right)+4\exp\left(-\frac{T}{\tau}\right)+2\frac{T}{\tau}\right]\sigma_{ou}^2 \tag{4.7}$$

$$q12 \;=\; \frac{\tau^2}{2}\left[1+\exp\left(-2\frac{T}{\tau}\right)-2\exp\left(-\frac{T}{\tau}\right)\right]\sigma_{ou}^2 \tag{4.8}$$

$$q22 \;=\; \frac{\tau}{2}\left[1-\exp\left(-2\frac{T}{\tau}\right)\right]\sigma_{ou}^2. \tag{4.9}$$

Stationary Gaussian White Noise Process Speed

The motivation for this model is to provide an analog to the previous model, but without the assumption of any correlation in the speed over time.

We once again assume that a target's speed fluctuates around some known mean value. However, in this case the deviations are serially uncorrelated.

The system is defined directly in discrete time with sample period T as:

$$\begin{bmatrix} x[t+T] \\ v[t+T] \end{bmatrix} \;=\; \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x[t] \\ v[t] \end{bmatrix} + \begin{bmatrix} T \\ 1 \end{bmatrix} u_v[t] + w[t] \tag{4.10}$$

where $u_v[t]$ is a control input that sets the mean speed value and $w[t] \sim N\left(0,\, Q_{sg}[t]\right)$ with

$$Q_{sg}[t] \;=\; \begin{bmatrix} T & 0 \\ 0 & 1 \end{bmatrix} \sigma_{sg}^2. \tag{4.11}$$

The speed model in this case is a Gaussian white noise process.

From the definition of the system we can see that the model makes a rather trivial estimate of the target's speed— that it is always at the mean (i.e. control) value, regardless of the current state. In this sense, it behaves like the previous system when the sample period is much larger than the correlation time.

4.3.2.2 Parameters

Suitable parameter values for each system are defined with respect to a particular motion regime. Here we restrict our attention to motion regimes defined solely based on a target's speed since that is what is relevant to the systems under consideration. Each such regime can be thought of as a distinct modality of a target's speed. For example, the motion of a human target whose position is sampled every 30 s might be described using "stop" and "walk" regimes. In some cases, that might be extended to include a "run" regime. Multiple "walk" regimes, each conditioned on the local terrain, could also be introduced.

Formally, we define a motion regime as when a target moves such that its speed can be regarded as wide-sense stationary. This implies that regimes can be characterized by their mean speed, m, and variance, q. We assume that an estimate of the speed distribution for the given motion regime is available or can be obtained using sufficiently accurate measurements of target proxies.

Returning to human target example, figure 4.1 shows that the distribution of speeds for steady-state walking. The mean and variance of the distribution are assumed to characterize the "walk" regime. To characterize the "stop" regime, we would have to analyze the speed measurements having filtered out the steady-state walking. This would lead to a peaked distribution of speeds with a mode at zero.

For the first and second systems, we assume that an estimate of the stationary autocorrelation coefficient function (ACF) is also available for each regime. In practice, this can be computed from the same speed time series that was used to estimate the speed distribution. Note that care must be taken to divide the complete speed time series into sub-series corresponding to each regime. All sub-series should be at least wsz seconds long, where the exact value is chosen experimentally to be just long enough to contain any statistically significant correlation. The ACF can then be estimated using a windowed approach, with window size wsz, such as Welch's method for spectral density estimation. In addition to noise reduction, this approach provides an elegant way to

combine multiple disjoint sub-series corresponding to the same motion regime.

Note that in cases where there is more than one motion regime, the different sets of parameters for a given system constitute different models and the subsequent estimation should proceed using a multiple model framework such as the interacting multiple model estimator (IMM) or generalized pseudo-Bayesian (GPB) estimator. Note that some special attention is required when working with more than one motion regime, which will be addressed later.

Let m and q be the mean and variance, respectively, of the motion regime for which the parameters are to be estimated.

Wiener Process Speed

This system contains a single parameter, σ_w, that controls the magnitude of the speed changes that a target undergoes over time.

To determine its value, we must first compute this magnitude for a target operating in the given motion regime.

Let the speed of such a target at times t_1 and t_2 (with $t_2 \geq t_1$) be defined as $S_1 = \text{speed}(t_1)$ and $S_2 = \text{speed}(t_2)$, respectively. The magnitude of the speed changes that it undergoes from time t_1 to t_2 is then given by the standard deviation of the difference between those two speeds, $\text{Std}[S_2 - S_1]$.

Since the target's speed is assumed to be wide-sense stationary, we get:

$$\begin{aligned} \text{Var}\left[S_2 - S_1\right] &= \text{Var}\left[S_1\right] + \text{Var}\left[S_2\right] - 2\text{Cov}\left[S_1,\, S_2\right] \\ &= q + q - 2qR\left(t_1,\, t_2\right) \\ &= 2q\left(1 - R\left(T\right)\right), \end{aligned}$$

where q is the stationary variance of the speed and $R\left(T\right)$ is the stationary ACF (an estimate of which was assumed earlier to be available) for lag $T = t_2 - t_1$.

From the discrete-time process noise covariance for this system we can see that over the sampling period T these changes are of the order $\sqrt{T}\sigma_w$.

Equating these two magnitudes we get:

$$\sqrt{T}\sigma_w = \sqrt{2q\left(1 - R\left(T\right)\right)}$$

which yields the parameter value

$$\sigma_w \quad = \quad \sqrt{\frac{2q\left(1 - R\left(T\right)\right)}{T}}. \tag{4.12}$$

When the stationary autocorrelation of a target's speed can be regarded as negligible, the parameter value becomes $\sqrt{\frac{2q}{T}}$. In practice, we expect this to be the case in applications where this model is used because the OU process model is likely to be a more suitable in the case of statistical significant autocorrelation in a target's speed.

Ornstein-Uhlenbeck Process Speed

This system contains three parameters: τ, σ_{ou}, and $u_v\left(t\right)$.

The first parameter, τ, is the relaxation time constant. It controls the rate at which the OU process relaxes to its mean value and characterizes the stationary ACF of the target speed, which is of the form:

$$R(L) = \exp\left(-\frac{|L|}{\tau}\right),$$

where L is the lag. Note that the decorrelation time is approximately 3τ (i.e. $R(3\tau) \approx 0.05$).

An appropriate value for τ can be determined by fitting the theoretical ACF to the empirical estimate (assumed earlier to be available).

The second parameter, σ_{ou}, controls the magnitude of the fluctuations in target speed around its mean value.

There are two approaches that we might use to determine its value. Both approaches rely on the assumption that a target's speed in the given motion regime is wide-sense stationary so the magnitude of these fluctuations is known to be of the order $\sqrt{q}$, where q is the stationary variance of the speed.

In the first approach, we equate the magnitude of the speed fluctuations to the stationary standard deviation of the OU speed model. That is to say we set σ_{ou} so that in the long term, after any correlation has died out, the steady-state variance of the OU speed model is q.

As $t \to \infty$, the OU process reverts to its stationary mean with variance $\sigma_{ou}^2 \frac{\tau}{2}$ and standard deviation $\sigma_{ou}\sqrt{\frac{\tau}{2}}$. This can be seen by looking at equation (4.9) and observing the variance as $T \to \infty$.

Equating these two magnitudes we get:

$$\sigma_{ou}\sqrt{\frac{\tau}{2}} \quad = \quad \sqrt{q}$$

which yields the parameter value

$$\sigma_{ou} \quad = \quad \sqrt{\frac{2q}{\tau}}. \tag{4.13}$$

In the second approach, we equate the magnitude of the speed fluctuations to the standard deviation of the OU speed model over the given sample period T. This ensures that the variance of the OU speed model is q regardless of how much correlation remains after T seconds.

The standard deviation for sample period T is given by equation (4.9).

Equating the two magnitudes we get:

$$\sigma_{ou}\sqrt{\frac{\tau}{2}\left[1-\exp\left(-2\frac{T}{\tau}\right)\right]} \quad = \quad \sqrt{q}$$

which yields the parameter value:

$$\sigma_{ou} \quad = \quad \frac{\sqrt{2q}}{\sqrt{\tau\left[1-\exp\left(-2\frac{T}{\tau}\right)\right]}}. \tag{4.14}$$

The third and final parameter, $u_v\,(t)$, is the control input that determines the mean value of the OU process. Its value expresses the postulated preferred speed for targets operating in the given motion regime. Thus, it takes the value m, the stationary mean of the speed.

Stationary Gaussian White Noise Process Speed

This system contains two parameters: σ_{sg} and $u_v\,(t)$.

The first parameter, σ_{sg}, controls the magnitude of the fluctuations in target speed around its mean value. Since we have assumed that the speed of a target operating in the given motion regime has a stationary variance of q and the speed model for this system is a stationary Gaussian white noise process with variance σ_{sg}^2, we see immediately that the parameter value is given by:

$$\sigma_{sg} \quad = \quad \sqrt{q}. \tag{4.15}$$

The second and final parameter, $u_v(t)$, is the control input that determines the mean value of the speed process. As in the previous system, its value expresses the postulated preferred speed for targets operating in the given motion regime and it takes the value m, the stationary mean of the speed.

4.3.2.3 Estimate Uncertainty

By design, the parameters have been selected such that variance in the estimated speed over the sampling period T is nominally equal to q.

For the first system, this is true when the stationary correlation in the speed is negligible; the typical case. For the second, this is true when σ_{ou} is set using the first approach and $T > 3\tau$ (i.e. the speed has decorrelated); it is always true when σ_{ou} is set using the second approach. It is always true for the third system.

To get better sense of how the systems differ, we will examine the variance in their position estimates.

Over the sampling period T, the variance in the estimated position from the first system is:

$$q\frac{2}{3}T^2\left(1 - R\left(T\right)\right). \tag{4.16}$$

For the second system, the position variance when σ_{ou} is set using the first approach is given by:

$$q\tau^2\left[4\exp\left(-\frac{T}{\tau}\right) - 3 - \exp\left(-2\frac{T}{\tau}\right) + 2\frac{T}{\tau}\right]. \tag{4.17}$$

and

$$\frac{q\tau^2\left[4\exp\left(-\frac{T}{\tau}\right) - 3 - \exp\left(-2\frac{T}{\tau}\right) + 2\frac{T}{\tau}\right]}{\left(1 - \exp\left(-2\frac{T}{\tau}\right)\right)}. \tag{4.18}$$

when σ_{ou} is set using the second approach.

Finally, for the third system, the position variance is Tq.

We see immediately that the level of uncertainty is substantially different in all three cases.

Since the variances in the position estimates from the first and third system are both functions only of T and q, we can compare them directly.

Using a stationary Gaussian white noise process to model the target speed introduces the system driving noise directly into the velocity. As expected, in the third system the position variance grows linearly with respect to time. When the system

driving noise is introduced directly into the acceleration, as in the case with the Wiener process speed model in the first system, the variance grows quadratically with respect to time.

Given these differences, do we have any reason to prefer one model over the other? Perhaps. When the empirical analysis of target speed suggests that it is well approximated using a stationary Gaussian white noise process, then the first system might be overstating the uncertainty in position for long sample periods. However, in situations where the assumption of a stationary mean target speed is a poor one, the first system will have an advantage since the Wiener process speed model is non-stationary.

Returning to the second system, let us look at what happens to the variance in the position estimate in the limits as $\tau \to 0$, and $\tau \to \infty$.

As $\tau \to 0$, the decorrelation time of the OU process gets shorter and we expect the modeled speed to become white noise. It is tempting to think then that in this limit the position variance should approach that of the third system.

However, we can see from equations (4.17) and (4.18) that this is not the case for either form of σ_{ou}. The limiting variance of the position estimate in both cases is in fact zero. But, for a given sample period, as τ approaches zero we are presumably integrating speed with less and less correlation and nonzero variance; the resulting uncertainty in position cannot be zero. This is obviously an unphysical result.

To understand what is going on, we need to take a closer look at the properties of the OU process.

According to the zero-tau limit theorem [67], the OU process becomes a Gaussian white noise process with variance ϵ in the limit as $\tau \to 0$ if and only if the product $\tau^2\sigma_{ou}^2 = \epsilon$ is constant with respect to τ.

This implies that if the variance of the white noise process reached in the limit is a (constant with respect to τ), then the driving noise must be of the form $\frac{a}{\tau^2}$. It follows that the stationary variance of such a process is $\frac{a}{2\tau}$. Clearly, there is no value for a which both satisfies our design constraints and produces the desired limiting position variance.

Our choices for σ_{ou} were based on matching the variance (stationary or instantaneous; depending on the approach) of the OU process speed model to the empirically derived stationary variance of target speed. Unfortunately, they both lead to driving noise variances that are functions of τ such that as $\tau \to 0$, the OU process becomes a

Gaussian white noise process with a variance that becomes zero in the limit.

Thus, for values of τ below some threshold the variance in the position estimate with be anomalously low and system two is an unsuitable choice. Under these circumstances system three should be preferred.

Before discussing how this threshold value is determined, let us continue the analysis to see what happens with large values of τ.

As $\tau \to \infty$, the decorrelation time of the OU process gets longer and we expect the modeled speed to become the integral of white noise– a Wiener process. In this limit the position variance should approach that of the first system, assuming that $R(T) = \exp(-\frac{T}{\tau})$, or

$$\lim_{\tau \to \infty} \; q\frac{2}{3}T^2 \left(1 - \exp\left(-\frac{T}{\tau}\right)\right) = 0. \tag{4.19}$$

In this case, arriving at zero uncertainty makes intuitive sense since, for a given sample period, as τ approaches infinity we are integrating speed with more and more correlation, hence speed with less and less variance.

When σ_{ou} is set using the first approach, it can be shown from equation (4.17) that the limiting variance is zero as expected. Since the limiting value in both (4.19) and the limit of (4.17) is dominated by $\exp\left(-\frac{T}{\tau}\right)$, the speed model does indeed converge to the Wiener process model.

When σ_{ou} is set using the second approach, it can be shown from equation (4.18) that the limiting variance is $\frac{1}{3}T^2q$. This is equivalent to the position variance from the first system assuming the magnitude of the fluctuations in target speed are half of the desired value and $R\,(T) = 0$.

Based on these results, the second approach to setting σ_{ou} should be preferred.

Figure 4.9 illustrates the behavior, discussed above, of the position uncertainty from system two as a function of τ for fixed (and arbitrary) values of q and T. It shows how the uncertainty from system two collapses to zero, failing to converge to match system three, as τ goes to zero and then converges to match the uncertainty from system one, with the appropriate correlation, as τ goes to infinity.

Note that changing the value of q in figure 4.9 would alter the values along the y-axis but not affect the relative relationship between the curves. However, it is important to point out that changing T would in fact alter the relative relationship between the curves.

For systems one and two, the value of T affects the rate of decay (it increases

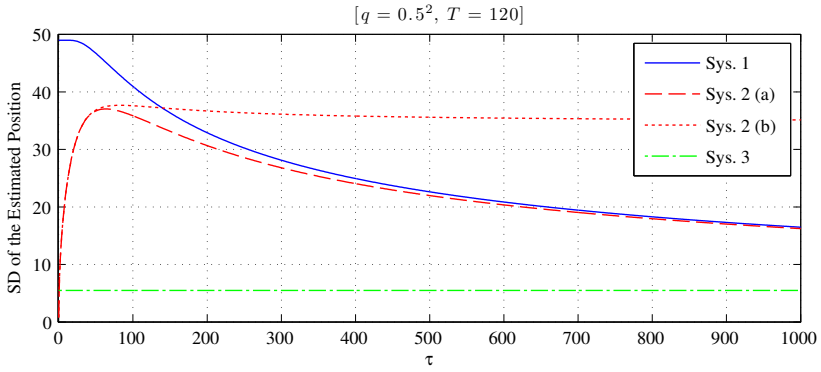


Figure 4.9: Standard deviation of the position estimate versus τ for fixed (and arbitrary) values of q and T.

for smaller sample periods) but leaves the relative relationship between the curves intact. For system three, the value of T affects where the curve lies along the y-axis.

Given a sufficiently small sample period, system three will produce the greatest uncertainty and the curve will lie above the others. As the size of the sample period increases, the curve for system three will drop below the peak uncertainty from system two.

This dependency on the sample period T in the relationship between the position uncertainty in systems two and three has implications for determining the minimum τ threshold that we have deferred until now.

A conservative choice for the threshold would be $\tau_{min} = \frac{T_{min}}{2}$, where T_{min} is the shortest sample period for the application. This places the threshold at the (approximate) location of the maximum position uncertainty from system two and would guarantee a physically sound result.

The choice is also logical if we adopt the view that as τ increases, the uncertainty over the given sample period should decrease; the same arguments used in the analysis of the limit $\tau \to \infty$ apply here.

Of course, when T_{min} is large enough that the position uncertainty from system three is less than the maximum uncertainty from system two, we might adopt a different view.

In this view, as τ goes from τ_{min} to infinity, the speed model is ideally changing from the stationary white noise model of system three to the Wiener process model (with exponential autocorrelation) of system one, and the estimate uncertainty should reflect

this transition. The threshold value would then be determined by setting equation (4.17) equal to Tq, with $T = T_{min}$, and solving for τ.

4.3.2.4 Multiple Models

In situations in which we have multiple motion regimes, some special attention is required to properly apply the different models. Specifically, care must be taken to ensure that when switching between motion regimes, any dependence of the current estimate on the previous speed of the target is properly accounted for. This is best illustrated by considering the typical case in which we have models for the "stopped" and "walk" regimes, although similar issues can arise when a target switches between any two different motions regimes.

For example, suppose that a target's speed was last estimated to be 1.3 m/s using a model for the "walk" regime and then during the next sample period the target abruptly stops. If we were to simply apply one of the speed models for the "stop" regime as stated above, then only the third system would be guaranteed to make a reasonable estimate. Since the previous speed is ignored by the model, the estimated speed would be near zero (taken directly from the parameters) and the estimated position approximately constant as we would expect for a stopped target. The second system might make a reasonable estimate depending on the sample period. A long enough sample period with respect to the correlation time and the influence of the current speed would be negligible. The first system would always predict a speed and position that are inconsistent with a stopped target by assuming that the target maintains its walking speed.

The solution is to handle transitions between motion regimes differently than consecutive estimates within a regime. When the motion regime associated with the previous estimate is different than the one being evaluated, the appropriate control inputs should be applied to force the estimated speed to the mean value for motion regime. The control inputs should be such that the position state is forced to values that are consistent with the set speed. This will ensure that when a target switches motion regimes between samples, the initial estimate will be a reasonable value. Such an adjustment is justified based on our assumption that the maneuver time for a target is much less than the sample period. The adjustment is thus intended to account for the unobserved behavior of the target.

This, of course, requires that we keep track of model used to generate the

current estimate. Some multiple model algorithms make a discrete choice which simply needs to be recorded. Usually, however, the current estimate is based on a mixture of the estimates from the active models. In this case, we can assume that the most likely model identifies the current motion regime.

Note that this idea of forcing the estimated speed to some model-determined value could also be used to create a variant of the first system that behaves a bit more like the third, in the sense that the speed does not drift over time. By applying control inputs at the time of each sample to effectively reset the previous speed to the mean value for the motion regime, the speed model of this new system becomes a stationary Gaussian process. However, it retains the process noise structure of the first system. This is because the new system is still based on the Wiener process speed model but simply resets the initial state of the process for each sample. Thus, the speed and position remain coupled through the acceleration. The resulting uncertainty in the position is larger than that of the third system reflecting the assumption of approximately constant speed, except for white acceleration noise, implied by the Wiener process model. This is arguably a more realistic than the uncorrelated white noise speed model.

4.3.3 Maneuver Modeling via System Rotation

Having looked at various ways in which the speed of the target can be modeled, we will now consider how to model the directional component of the target's motion.

We begin by extending the state space to describe the 2-D motion of a target in a plane tangential to the earth's surface. This can be done by replicating the earlier 1-D definition so that the state vector consists of the 2-D Cartesian position and the corresponding Cartesian velocity components: $[\ x \quad y \quad v_x \quad v_y\ ]^\top$. A representation of the velocity using polar coordinates will be considered in a subsequent section.

Recall that under our initial assumptions, within a single sample period targets can change to any heading. The assumptions further stipulate that this maneuverability is limited primarily by environmental factors or target intent. For now, let us put aside the notion of these high level constraints and look at the unconstrained case as it provides the motivation for this class of models. We will return later to the issue of environmental and intent-based constraints.

Without a preferred heading, we can regard as equally likely all target state predictions that differ only in rotation about the current position. Unfortunately, it is

impossible to accurately represent such a belief using a single linear Gaussian kinematic model because the distribution of positions would require a ring-like shape. However, it can be approximated—in theory—using multiple models. Given a single kinematic base model and its prediction, we could postulate a set of models whose predictions cover all possible rotations of the base prediction about the current target position. Of course, this would be completely impractical due to the number of models that would be required. Even a reasonably coarse discretization of the headings would still be impractical.

An alternative approach that goes a long way toward achieving the desired result is to use a single model based on an estimate of the optimal rotation. The idea being that the optimal rotation corresponds to the model with the highest probability of being correct. A simple and effective approximation to the optimal rotation is the one that points the predicted velocity toward the current measurement. This rotation value could be calculated with or without taking into account the measurement uncertainty, depending on the computational resources that are available.

Clearly, this involves some abuse of the notion that the model should predict the future state of the target based on its present state. We justify it as an approximation to a proper treatment, using multiple models, of equally probable changes in direction.

Let θ be the angle between the vector $\begin{bmatrix} v_x[t+T] & v_y[t+T] \end{bmatrix}^\top$ and a vector representing the optimal rotation.

The desired model is then given by:

$$\begin{bmatrix} x[t+T] \\ y[t+T] \\ v_x[t+T] \\ v_y[t+T] \end{bmatrix} = FR_\theta \begin{bmatrix} x[t] \\ y[t] \\ v_x[t] \\ v_x[t] \end{bmatrix} + BR_\theta u[t] + w[t],$$

where $w[t] \sim \mathcal{N}(0,\, Q_\theta)$ with

$$Q_\theta = \begin{bmatrix} q11 & q12 & q13 & q14 \\ q12 & q22 & q23 & q24 \\ q13 & q23 & q33 & q34 \\ q14 & q24 & q34 & q44 \end{bmatrix}^{-1} \tag{4.20}$$

and

$$
\begin{aligned}
q11 &= Q_{1,1}^{-1} \\
q12 &= Q_{1,2}^{-1} \\
q13 &= Q_{1,3}^{-1}\cos(\theta) - Q_{1,4}^{-1}\sin(\theta) \\
q14 &= Q_{1,4}^{-1}\cos(\theta) + Q_{1,3}^{-1}\sin(\theta) \\
q22 &= Q_{2,2}^{-1} \\
q23 &= Q_{2,3}^{-1}\cos(\theta) - Q_{2,4}^{-1}\sin(\theta) \\
q24 &= Q_{2,4}^{-1}\cos(\theta) + Q_{2,3}^{-1}\sin(\theta) \\
q33 &= Q_{3,3}^{-1}\cos(\theta)^2 - 2Q_{3,4}^{-1}\cos(\theta)\sin(\theta) + Q_{4,4}^{-1}\sin(\theta)^2 \\
q34 &= Q_{3,4}^{-1}\cos(2\theta) + \frac{1}{2}\left(Q_{3,3}^{-1} - Q_{4,4}^{-1}\right)\sin(2\theta) \\
q44 &= Q_{4,4}^{-1}\cos(\theta)^2 + 2Q_{3,4}^{-1}\cos(\theta)\sin(\theta) + Q_{3,3}^{-1}\sin(\theta)^2,
\end{aligned}
$$

where Q is the process noise covariance matrix of the base model. Note that the rotated process noise covariance, Q_θ, is defined in terms of its inverse. The terms $Q_{m,n}^{-1}$ refer to elements in the inverse of the base model process noise covariance. F and B specify the system matrix and input gain matrix of the base model, respectively. R_θ is the matrix used to rotate the system and input gain matrices, which is given by:

$$
R_\theta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\theta) & -\sin(\theta) \\ 0 & 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}. \tag{4.21}
$$

When environmental or intent-based constraints dictate a preferred heading, the optimal rotation can be defined based those without regard to the measurement. Of course, when the sample period, T, is large, the accuracy of any prediction based on local constraints is likely to suffer.

4.3.4 Position-Speed-Heading

Here, we look at a different approach to modeling the directional component of the target's 2-D motion. It is based on decomposing the velocity into a magnitude and direction. Thus, the Cartesian velocity components used in the previous section are

replaced with the target's speed (i.e. absolute value of the velocity) and heading angle to get the state vector: $[\ x \quad y \quad v \quad h\ ]^\top$.

This polar representation of the target's velocity is convenient because it enables us to think about, and model, the speed and direction of travel separately. However, it comes at a cost since the relationship of the speed and heading to the Cartesian position is nonlinear. This means that we cannot discretize the continuous-time dynamics using the the straightforward techniques from linear system theory.

The usual way forward in such cases is to first linearize the system via a Taylor series expansion of the dynamics equations and then discretize the resulting linear approximation. This can be referred to as the *discretized linearization* approach, which is the basis for the well-known Extended Kalman Filter (EKF). However, this is not the only possible approach. Alternatively, we can discretize the system via integration first and then linearize the resulting discrete-time system using a Taylor series expansion. This is the so-called *linearized discretization* approach.

In the context of coordinated turn tracking, Gustafsson and Isaksson [76] studied these two different approaches and concluded that the later is preferable because the total linearization error was less. Their work also lends support to the use of polar velocity as they show analytically that for coordinated turns it leads to a smaller linearization error. For aircraft tracking, Gertz [65] reports good performance using a filter derived by discretizing prior to linearization, although no comparison is made to the traditional approach nor is any special justification offered.

We will consider both approaches despite the fact that, when using a Taylor series expansion to linearize the system, the results may be identical in some cases. One such case is the current system given Wiener process models for the speed and heading. The reason for continuing to consider discretization first is that the Taylor series expansion is not the only way to complete the approximation. The unscented transform (UT) provides yet another approach to linearizing a nonlinear system [91, 92]. As noted in previous chapters, the UT has proven to be a more effective and less error-prone alternative to a Taylor series expansion. However, it does not get us around the problem of discretizing the nonlinear system model. What it does do is give us another more attractive option for the linearization step in the linearized discretization approach. It will also turn out to be useful in a novel method for estimating the discrete process noise covariance for nonlinear systems.

Let us assume that the target's speed and heading are constant except for stochastic variability in the form of Gaussian white noise in their rates of change. Thus, the speed and heading are modeled using independent Wiener processes. The continuous-time dynamic model is then given by:

$$\begin{aligned} \dot{x}(t) &= v(t)\cos\left(h(t)\right) \\ \dot{y}(t) &= v(t)\sin\left(h(t)\right) \\ \dot{v}(t) &= w_v(t) \\ \dot{h}(t) &= w_h(t), \end{aligned} \tag{4.22}$$

where $w_v \sim \mathcal{N}(0,\, \sigma_v^2)$ and $w_h \sim \mathcal{N}(0,\, \sigma_h^2)$. The noise processes driving the changes in the speed and heading are assumed here to be uncorrelated for simplicity.

4.3.4.1 Discretized Linearization

Discretized linearization is the familiar approach in which the system is first linearized and then the linear approximation is discretized using the techniques from linear system theory.

Let the continuous-time dynamic model be written as:

$$\dot{s}(t) = f(s(t)) + w(t),$$

where $s(t) = \begin{bmatrix} x(t) & y(t) & v(t) & h(t) \end{bmatrix}^\top$ is the vector representing the instantaneous state of the target, $f(s(t))$ is a vector-valued state evolution function, and $w(t)$ is the noise driving the system. The noise is distributed as $w(t) \sim N\left(0,\, Q(t)\right)$ with

$$Q(t) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & \sigma_h^2 \end{bmatrix}. \tag{4.23}$$

A first order vector Taylor series expansion around the current state estimate $\hat{s}$ gives us:

$$\dot{s}(t) \approx f(\hat{s}) + f'(\hat{s})\left(s(t) - \hat{s}\right) + w(t),$$

where f' denotes the Jacobian matrix $\mathbf{J}_f(x,\, y,\, v,\, h)$.

From this we can derive the following discrete-time linear approximation:

$$s[t+T] \approx F[t]s[t] + G[t]u[t] + w[t],$$

where

$$\begin{aligned} F[t] &= \exp\left(f'(\hat{s})\,T\right) \\ G[t] &= \int_0^T \exp\left(f'(\hat{s})\,\tau\right) d\tau \\ u[t] &= f(\hat{s}) - f'(\hat{s})\,\hat{s} \\ w[t] &= \int_0^T \exp\left(f'(\hat{s})\,\tau\right) w(\tau) d\tau. \end{aligned} \tag{4.24}$$

The discrete-time process noise, $w[t]$, is zero-mean Gaussian with covariance:

$$Q[t] = \int_0^T \exp\left(f'(\hat{s})\,\tau\right) Q(t) \exp\left(f'(\hat{s})\,\tau\right)^\top d\tau. \tag{4.25}$$

Note that the last $\top$ in the integrand above indicates that the transpose of the matrix exponential; it is not to be confused with the sample period.

For the system defined in (4.22) with current state $\hat{s} = \begin{bmatrix} \hat{x} & \hat{y} & \hat{v} & \hat{h} \end{bmatrix}^\top$, the Jacobian matrix is:

$$f'(\hat{s}) = \begin{bmatrix} 0 & 0 & \cos\left(\hat{h}\right) & -\hat{v}\sin\left(\hat{h}\right) \\ 0 & 0 & \sin\left(\hat{h}\right) & \hat{v}\cos\left(\hat{h}\right) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

which gives us the following discrete-time linear approximation:

$$F[t] = \begin{bmatrix} 1 & 0 & T\cos\left(\hat{h}\right) & -T\hat{v}\sin\left(\hat{h}\right) \\ 0 & 1 & T\sin\left(\hat{h}\right) & T\hat{v}\cos\left(\hat{h}\right) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.26}$$

$$G[t] = \begin{bmatrix} T & 0 & \frac{1}{2}T^2\cos\left(\hat{h}\right) & -\frac{1}{2}T^2\hat{v}\sin\left(\hat{h}\right) \\ 0 & T & \frac{1}{2}T^2\sin\left(\hat{h}\right) & \frac{1}{2}T^2\hat{v}\cos\left(\hat{h}\right) \\ 0 & 0 & T & 0 \\ 0 & 0 & 0 & T \end{bmatrix}, \tag{4.27}$$

and

$$u[t] = \begin{bmatrix} \hat{h}\hat{v}\sin\left(\hat{h}\right) & -\hat{h}\hat{v}\cos\left(\hat{h}\right) & 0 & 0 \end{bmatrix}^\top. \tag{4.28}$$

The discrete-time process noise covariance is given by:

$$Q[t] = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} & Q_{14} \\ Q_{12} & Q_{22} & Q_{23} & Q_{24} \\ Q_{13} & Q_{23} & Q_{33} & 0 \\ Q_{14} & Q_{24} & 0 & Q_{44} \end{bmatrix}, \tag{4.29}$$

where

$$\begin{aligned}
Q_{11} &= \frac{1}{3}T^3\left(\sigma_v^2\cos\left(\hat{h}\right)^2 + \sigma_h^2\hat{v}^2\sin\left(\hat{h}\right)^2\right) \\
Q_{12} &= \frac{1}{6}T^3\left(\sigma_v^2 - \sigma_h^2\hat{v}^2\right)\sin\left(2\hat{h}\right) \\
Q_{13} &= \frac{1}{2}T^2\sigma_v^2\cos\left(\hat{h}\right) \\
Q_{14} &= -\frac{1}{2}T^2\sigma_h^2\hat{v}\sin\left(\hat{h}\right) \\
Q_{22} &= \frac{1}{3}T^3\left(\sigma_h^2\hat{v}^2\cos\left(\hat{h}\right)^2 + \sigma_v^2\sin\left(\hat{h}\right)^2\right) \\
Q_{23} &= \frac{1}{2}T^2\sigma_v^2\sin\left(\hat{h}\right) \\
Q_{24} &= \frac{1}{2}T^2\sigma_h^2\hat{v}\cos\left(\hat{h}\right) \\
Q_{33} &= T\sigma_v^2 \\
Q_{44} &= T\sigma_h^2.
\end{aligned}$$

4.3.4.2 Linearized Discretization

Linearized discretization is the alternative approach in which the system is first discretized and then linearized.

The nonlinear state transition function, without noise, can be discretized via integration by:

$$s[t+T] = s[t] + \int_t^{t+T} f(s(\tau))\,d\tau.$$

Once again, square brackets have been used to denote discrete-time elements to avoid confusion. Discretization of the process noise will be addressed later.

Expanding the nonlinear state transition function, for the system defined in

(4.22), yields the following discrete-time state update equations:

$$\begin{aligned} x[t+T] &= x[t] + \int_t^{t+T} v(\tau)\cos\left(h(\tau)\right) d\tau \\ y[t+T] &= y[t] + \int_t^{t+T} v(\tau)\sin\left(h(\tau)\right) d\tau \\ v[t+T] &= v[t] \\ h[t+T] &= h\left[t\right]. \end{aligned}$$

Combining the individual state update equations into a vector-valued function, the discrete-time nonlinear system, still without noise, can be written as:

$$s[t+T] = g(s[t];\, T)\,, \tag{4.30}$$

where the dependence of the state update function on the sample period, T, has been made explicit.

A first order vector Taylor series expansion around the current state estimate $\hat{s}$ gives us:

$$s[t+T] \approx g(\hat{s}) + g'(\hat{s})\left(s[t] - \hat{s}\right),$$

where g' denotes the Jacobian matrix $\mathbf{J}_g\left(x,\, y,\, v,\, h\right)$.

From this we can derive the following discrete-time linear approximation:

$$s[t+T] \approx F[t]s[t] + G[t]u[t] + w[t],$$

where

$$\begin{aligned} F[t] &= g'(\hat{s}) \\ G[t] &= \mathbf{I}_4 \\ u[t] &= g(\hat{s}) - g'(\hat{s})\,\hat{s} \\ &= g(\hat{s}) - F[t]\hat{s}. \end{aligned} \tag{4.31}$$

The noise term has been added here for the sake of completeness; it will be discussed below.

For the system defined in (4.22) with current state $\hat{s} = \left[\begin{array}{cccc} \hat{x} & \hat{y} & \hat{v} & \hat{h} \end{array}\right]^\top$, the resulting discrete-time linear approximation has the same form as the approximation that we derived earlier using the discretized linearization approach. The system matrix, $F[t]$, is given by (4.26) and the control input, $u[t]$, is given by the product of (4.27) and (4.28). This is because the deterministic portions of the derivatives, with respect to

time, of the speed and heading are constant. Thus, their integrals over the sample period have a trivial form. This will not be the case when we consider a position-speed-heading system with exponentially autocorrelated noise later.

The UT can also be used to linearize the system around the current state estimate.

Given $x \sim \mathcal{N}(\mu_x, \Sigma_x)$ and $g(\cdot)$, the UT provides a Gaussian approximation for the joint distribution of x and y of the form [81]:

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{xy}^\top & \Sigma_y \end{bmatrix} \right).$$

Letting x represent the state of the target at time t and y the state at time $t+T$, we can derive the following discrete-time linear approximation:

$$s[t+T] \approx F[t]s[t] + G[t]u[t] + w[t],$$

where

$$\begin{aligned} F[t] &= \Sigma_{xy}^\top \Sigma_x^{-1} \\ G[t] &= \mathbf{I}_4 \\ u[t] &= \mu_y - \Sigma_{xy}^\top \Sigma_x^{-1} \mu_x \\ &= \mu_y - F[t]\mu_x. \end{aligned} \tag{4.32}$$

The noise term has again been added, but this time because the linear approximation includes a contribution to the covariance. The discrete-time process noise covariance in this case is given by:

$$\begin{aligned} Q[t] &= \Sigma_y - \Sigma_{xy}^\top \Sigma_x^{-1} \Sigma_{xy} + Q_{nl}[t] \\ &= \Sigma_y - F[t]\Sigma_x^{-1} F[t]^\top + Q_{nl}[t], \end{aligned} \tag{4.33}$$

where $Q_{nl}[t]$ is the process noise covariance of the nonlinear system, which we will discuss next. Notice that the second term cancels out the contribution from the uncertain state propagating through the linearized system which is then replaced by the UT estimate of that covariance. Note that in practice, care must be taken to ensure that these calculations produce a positive semi-definite covariance. Unfortunately, the UT calculations do not guarantee that Σ_y is positive semi-definite. Checking the result and applying Tikhonov regularization, if necessary, is the easiest way to ensure numerical stability.

Let us now consider how to discretize the process noise to complete the definitions of our discrete-time linear approximations.

Unfortunately, there are no universally accepted techniques for directly discretizing (i.e. without linearization) the noise in nonlinear systems to determine an expression for $Q[t]$. Gustafsson and Isaksson [76] identify five different alternatives for computing the discrete-time process noise covariance and consider two of them in the context of linearized discretization. Those are:

$$Q_a[t] = TQ(t) \tag{4.34}$$

$$Q_b[t] = TF[t]Q(t)F[t]^\top. \tag{4.35}$$

Note that the last $\top$ in the equation for $Q_b[t]$ above indicates the transpose of the system matrix; it is not to be confused with the sample period.

Although Gustafsson and Isaksson reported that (4.34) performed at least as well as (4.35) in their evaluation, it is unclear exactly how they reached that conclusion. The former is an extremely crude approximation that fails to account for the propagation of the driving noise through the system so there will be zero uncertainty in the position estimates. In our experience, the latter should be preferred.

The other three alternatives were considered in the context of discretized linearization. The standard method for noise discretization in linear systems, based on the continuous integration of $w(t)$, which we used to define $Q[t]$ in the previous section, is included among the three.

In addition to all these possibilities, we will propose two new approaches to computing the discrete-time noise covariance based on the integration of $w(t)$.

The first approach is to use the discrete-time linear approximation to the system matrix, $g'(\hat{s})$, and the standard method for linear systems to get:

$$Q_c[t] = \int_0^T F[t;\, \tau]Q(t)F[t;\, \tau]^\top d\tau \tag{4.36}$$

where $F[t;\, \tau]$ indicates the system matrix for the sample period τ. Note that the last $\top$ in the integrand indicates the transpose of the system matrix; it is not be confused with the sample period.

For the current system, the resulting covariance is the same as what we derived earlier using the discretized linearization approach, which is given by (4.29). However, that will not always be the case.

When the matrix, $F[t]$, derived in this section does in fact model the true system more accurately than the one derived in the previous section using discretized linearization, we can expect this approach to yield a more accurate representation of $Q[t]$. Compared to $Q_b[t]$, the covariance $Q_c[t]$ for the current system is smaller; by a factor of 3 for the block corresponding to the position and a factor of 2 in the cross terms for the speed and heading. This difference is due to the fact that the terms in the system matrix, that are functions of the sample period, are integrated rather than simply scaled. In our evaluations comparing the estimates to the covariance computed by simulating the continuous-time system, $Q_c[t]$ performs reasonably well, producing values within a few percent of the simulated values, while $Q_b[t]$ overstates the covariance in the position block as expected.

Although it is sometimes possible to derive a closed-form solution to this integral, the result can often be unwieldy. In such cases, we can use numerical methods to approximate the integral. When performance is critical, application-specific optimizations such as precomputing polynomial approximations to the integral can be employed.

The second approach to noise discretization extends the notion of continuous integration of $w(t)$ directly to nonlinear systems without requiring a linearization step. It works as follows.

First, recall that for a linear system, the discrete-time process noise can be written as:

$$w[t] = \int_0^T F[t;\, \tau] w(\tau) d\tau$$

where $F[t;\, \tau]$ again indicates the system matrix for a sample period of τ. In other words, $w[t]$ is the integral of the continuous-time process noise propagated through the discretized system, with a sample period given by the variable of integration. The propagation of the noise in this case is simply a linear transformation, so the covariance of $w[t]$ can be written as in (4.36).

From this we can see that the covariance, $Q[t]$, is the integral of the covariance of the continuous-time process noise propagated through the discretized system, with a sample period given by the variable of integration. In general, we can state the integral as:

$$Q[t] = \int_0^T \mathrm{Cov}(g(w(\tau);\, \tau))\, d\tau$$

where $g(w(\tau);\, \tau)$ is a vector-valued state update function, not necessarily linear, with current state $w(\tau)$ and sample period τ.

When $g(\cdot)$ represents a nonlinear transformation, the resulting covariance does not generally have a closed-form solution [110]. However, we can use the UT to efficiently approximate the covariance in this case.

Let the mean of x be equal to the current state estimate and the covariance be that of the continuous-time dynamic model. This gives us $\mu_x = \hat{s}$ and

$$\Sigma_x \quad = \quad \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & \sigma_h^2 \end{bmatrix}.$$

Let the nonlinear transformation relating x and y be given by (4.30), the vector-valued function representing the deterministic portion of the discrete-time system. Note that $g(\cdot)$ should be thought of as a function of both the state and the sample period used to compute it. Thus, y is also a function of the sample period.

The discrete-time process noise covariance is then given by:

$$Q_d[t] = \int_0^T \Sigma_{y|\tau}\, d\tau \tag{4.37}$$

where $\Sigma_{y|\tau}$ is the covariance matrix resulting from the UT; the dependency on the sample period, τ, is explicitly shown.

In principle, it is possible to derive a closed-form solution to this integral, however, it is usually not worth the effort. In practice, we prefer to use numerical methods to approximate it. We have found that using the trapezoidal rule with a coarse grid provides sufficient accuracy for this application. The properties of the driving noise can be exploited to make the numerical integration more efficient.

It is often the case that the dynamics of the noise that drives the system are linear. This enables us to appeal to linear systems theory to derive discrete-time covariance of those states. These values can then be used to verify the correspond elements of the approximated $Q[t]$.

For the current system, the noise properties are such that we can evaluate (4.37) using a single invocation of the UT at $\tau = T$. The resulting discrete-time noise

covariance is given by:

$$Q_d[t] = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} & Q_{14} \\ Q_{12} & Q_{22} & Q_{23} & Q_{24} \\ Q_{13} & Q_{23} & Q_{33} & 0 \\ Q_{14} & Q_{24} & 0 & Q_{44} \end{bmatrix} \tag{4.38}$$

where

$$\begin{aligned} Q_{11} &= \frac{1}{3}T\left(\Sigma_{y|\tau}\right)_{11} \\ Q_{12} &= \frac{1}{3}T\left(\Sigma_{y|\tau}\right)_{12} \\ Q_{13} &= \frac{1}{2}T\left(\Sigma_{y|\tau}\right)_{13} \\ Q_{14} &= \frac{1}{2}T\left(\Sigma_{y|\tau}\right)_{14} \\ Q_{22} &= \frac{1}{3}T\left(\Sigma_{y|\tau}\right)_{22} \\ Q_{23} &= \frac{1}{2}T\left(\Sigma_{y|\tau}\right)_{23} \\ Q_{24} &= \frac{1}{2}T\left(\Sigma_{y|\tau}\right)_{24} \\ Q_{33} &= T\left(\Sigma_{y|\tau}\right)_{33} \\ Q_{44} &= T\left(\Sigma_{y|\tau}\right)_{44}. \end{aligned}$$

4.3.4.3 Control Inputs

In the previous section on speed models, we noted that the ability to force the system to a specific speed, by applying a deterministic control input, is necessary when the model is selected after having previously used another. This ensures that the estimated speed at the end of the sample period is consistent with the assumptions used to determine the model parameters. It might also be desirable to incorporate the notion of the preferred target heading that was discussed earlier in the context of maneuver modeling via system rotation. This would require the ability to force the system to a specific heading by applying a deterministic control input.

For the system defined in (4.22), support for this functionality is defined as follows.

Let $\tilde{u}_v[t]$ be the speed control input which is applied during the sample interval $[t,\, t+T]$. It represents the desired speed of the target at time $t+T$. Similarly, let $\tilde{u}_h[t]$

be the heading control input that represents the desired direction of travel at time time $t + T$. The tilde notation is meant to signify that these are explicit control values intended to modify the state in a prescribed manner as opposed to the implicit control vector, $G[t]u[t]$, that represents the bias around the linearization.

For the Taylor series expansions, the linearization point is no longer the current state estimate at time t but rather the current position with the commanded speed and heading values. Thus, $\hat{s} = \begin{bmatrix} x[t] & y[t] & \tilde{u}_v[t] & \tilde{u}_h[t] \end{bmatrix}^\top$. The implicit control vector, $G[t]u[t]$, is computed as described above. However, it must be then be offset to account for the impact of the explicit control. The offset is given by:

$$\begin{bmatrix} -F_{13}\left(v[t] - \tilde{u}_v[t]\right) - F_{14}\left(h[t] - \tilde{u}_h[t]\right) \\ -F_{23}\left(v[t] - \tilde{u}_v[t]\right) - F_{24}\left(h[t] - \tilde{u}_h[t]\right) \\ -F_{33}\left(v[t] - \tilde{u}_v[t]\right) - F_{34}\left(h[t] - \tilde{u}_h[t]\right) \\ -F_{43}\left(v[t] - \tilde{u}_v[t]\right) - F_{44}\left(h[t] - \tilde{u}_h[t]\right) \end{bmatrix},$$

where F_{ij} denotes an element in $F[t]$.

When the UT is used to generate the approximation, the changes to the speed and heading values can be absorbed into $g(\cdot)$ by simply setting the speed and heading states to their commanded values prior to evaluating the original function.

As mentioned earlier, in addition to producing the expected results when working with multiple models, the speed control input can be used to create a stationary variant of the speed model that retains the original noise model. The behavior of an exponentially correlated speed model can also be approximated by setting the control input to some speed between the current and the desired, based on the sample period.

4.3.5 Position-Speed-Heading with Exponential Correlation

A polar representation of the velocity can also be used with exponentially correlated models. In other words, we can replace the Wiener process speed and heading models used in the previous section with Ornstein-Uhlenbeck process models.

However, before defining the continuous-time dynamic model we will first make a small change to the state space in order to simplify the definition of the model. Let the state vector be: $\begin{bmatrix} x & y & n_v & n_h \end{bmatrix}^\top$, where n_v and n_h are the speed noise and heading noise, respectively. The noise variables represent the difference between the instantaneous and desired values, which are specified as control inputs to the system.

Let $\tilde{u}_v$ and $\tilde{u}_h$ denote the desired speed and heading, respectively. The tilde notation is meant to signify that these are explicit control values intended to modify the state in a prescribed manner as opposed to the implicit control vector that represents the bias around the linearization. The mathematical form of the noise models, Uhlenbeck-Ornstein processes, implies that they have bounded variance and will drift towards zero, the equilibrium level of the process. Thus, the actual speed and heading of the target are given by $\tilde{u}_v + n_v$ and $\tilde{u}_h + n_h$, respectively.

The continuous-time dynamic model is given by:

$$\begin{aligned} \dot{x}(t) &= (\tilde{u}_v(t) + n_v(t)) \cos(\tilde{u}_h(t) + n_h(t)) \\ \dot{y}(t) &= (\tilde{u}_v(t) + n_v(t)) \sin(\tilde{u}_h(t) + n_h(t)) \\ \dot{n_v}(t) &= -\alpha_v n_v(t) + w_v(t) \\ \dot{n_h}(t) &= -\alpha_h n_h(t) + w_h(t) \end{aligned} \tag{4.39}$$

where α_v and α_h are the reciprocals of the correlation time for the speed and heading, respectively. The driving noise is $w_v \sim \mathcal{N}(0, \sigma_v^2)$ and $w_h \sim \mathcal{N}(0, \sigma_h^2)$.

As we did in the previous section, we will consider both the discretized linearization and linearized discretization approaches to generating a discrete-time linear approximation to the model.

4.3.5.1 Discretized Linearization

Recall that discretized linearization is the familiar approach in which the system is first linearized and then the linear approximation is discretized using the techniques from linear system theory.

Let the continuous-time dynamic model be written as:

$$\dot{s}(t) = f(s(t);\, \tilde{u}_v(t),\, \tilde{u}_h(t)) + w(t),$$

where $s(t) = [\ x(t) \quad y(t) \quad n_v(t) \quad n_h(t)\ ]^\top$ is the state vector, $f(s(t);\, \tilde{u}_v(t),\, \tilde{u}_h(t))$ is a vector-valued state transition function, which includes the explicit control inputs, and $w(t)$ is the noise driving the system. The noise is distributed as $w(t) \sim N(0,\, Q(t))$ with

$$Q(t) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & \sigma_h^2 \end{bmatrix}. \tag{4.40}$$

As we saw in section 4.3.4.1, the discrete-time linear approximation has the form:

$$s[t+T] \approx F[t]s[t] + G[t]u[t] + w[t],$$

where $w[t] \sim N(0, Q[t])$. The components of the approximation are described in (4.24) and (4.25).

For the system defined in (4.39) with current state estimate $\hat{s} = \begin{bmatrix} \hat{x} & \hat{y} & \hat{n}_v & \hat{n}_h \end{bmatrix}^\top$ and current control inputs $\hat{u}_v$ and $\hat{u}_h$, the Jacobian matrix is:

$$f'(\hat{s}) = \begin{bmatrix} 0 & 0 & \cos(\hat{u}_h + \hat{n}_h) & -(\hat{u}_v + \hat{n}_v)\sin(\hat{u}_h + \hat{n}_h) \\ 0 & 0 & \sin(\hat{u}_h + \hat{n}_h) & (\hat{u}_v + \hat{n}_v)\cos(\hat{u}_h + \hat{n}_h) \\ 0 & 0 & -\alpha_v & 0 \\ 0 & 0 & 0 & -\alpha_h \end{bmatrix},$$

which gives us the following discrete-time linear approximation:

$$F[t] = \begin{bmatrix} 1 & 0 & \frac{(1-\exp(-\alpha_v T))\cos(\hat{u}_h+\hat{n}_h)}{\alpha_v} & -\frac{(1-\exp(-\alpha_h T))(\hat{u}_v+\hat{n}_v)\sin(\hat{u}_h+\hat{n}_h)}{\alpha_h} \\ 0 & 1 & \frac{(1-\exp(-\alpha_v T))\sin(\hat{u}_h+\hat{n}_h)}{\alpha_v} & \frac{(1-\exp(-\alpha_h T))(\hat{u}_v+\hat{n}_v)\cos(\hat{u}_h+\hat{n}_h)}{\alpha_h} \\ 0 & 0 & \exp(-\alpha_v T) & 0 \\ 0 & 0 & 0 & \exp(-\alpha_h T) \end{bmatrix}, \quad (4.41)$$

$$G[t] = \begin{bmatrix} T & 0 & G_{13} & G_{14} \\ 0 & T & G_{23} & G_{24} \\ 0 & 0 & \frac{1-\exp(-\alpha_v T)}{\alpha_v} & 0 \\ 0 & 0 & 0 & \frac{1-\exp(-\alpha_h T)}{\alpha_h} \end{bmatrix}, \quad (4.42)$$

where

$$\begin{aligned} G_{13} &= \frac{\exp(-\alpha_v T)\left(1 + \exp(\alpha_v T)(\alpha_v T - 1)\right)\cos(\hat{u}_h + \hat{n}_h)}{\alpha_v^2} \\ G_{14} &= -\frac{\left(\frac{\exp(-\alpha_h T)-1}{\alpha_h} + T\right)(\hat{u}_v + \hat{n}_v)\sin(\hat{u}_h + \hat{n}_h)}{\alpha_h} \\ G_{23} &= \frac{\exp(-\alpha_v T)\left(1 + \exp(\alpha_v T)(\alpha_v T - 1)\right)\sin(\hat{u}_h + \hat{n}_h)}{\alpha_v^2} \\ G_{24} &= -\frac{\left(\frac{\exp(-\alpha_h T)-1}{\alpha_h} + T\right)(\hat{u}_v + \hat{n}_v)\cos(\hat{u}_h + \hat{n}_h)}{\alpha_h} \end{aligned}$$

and

$$u[t] = \begin{bmatrix} (\hat{n}_v + \hat{u}_v - \hat{v})\cos(\hat{n}_h + \hat{u}_h) + \hat{h}\,(\hat{n}_v + \hat{u}_v)\sin(\hat{n}_h + \hat{u}_h) \\ (\hat{n}_v + \hat{u}_v - \hat{v})\sin(\hat{n}_h + \hat{u}_h) - \hat{h}\,(\hat{n}_v + \hat{u}_v)\cos(\hat{n}_h + \hat{u}_h) \\ (\hat{v} - \hat{n}_v)\alpha_v \\ (\hat{h} - \hat{n}_h)\alpha_h \end{bmatrix}. \tag{4.43}$$

The discrete-time process noise covariance is given by:

$$Q[t] = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} & Q_{14} \\ Q_{12} & Q_{22} & Q_{23} & Q_{24} \\ Q_{13} & Q_{23} & Q_{33} & 0 \\ Q_{14} & Q_{24} & 0 & Q_{44} \end{bmatrix} \tag{4.44}$$

where

$$\begin{aligned} Q_{11} &= -\frac{\sigma_v^2 c\cos(b)^2}{2\alpha_v^3} - \frac{\sigma_h^2 da^2\sin(b)^2}{2\alpha_h^3} \\ Q_{12} &= \frac{1}{2}\left(\frac{-\sigma_v^2 c}{\alpha_v^3} + \frac{\sigma_h^2 da^2}{\alpha_h^3}\right)\cos(b)\sin(b) \\ Q_{13} &= \frac{\sigma_v^2 e\cos(b)}{2\alpha_v^2} \\ Q_{14} &= -\frac{\sigma_h^2 fa\sin(b)}{2\alpha_h^2} \\ Q_{22} &= -\frac{\sigma_v^2 c\sin(b)^2}{2\alpha_v^3} - \frac{\sigma_h^2 da^2\cos(b)^2}{2\alpha_h^3} \\ Q_{23} &= \frac{\sigma_v^2 e\sin(b)}{2\alpha_v^2} \\ Q_{24} &= \frac{\sigma_h^2 fa\cos(b)}{2\alpha_h^2} \\ Q_{33} &= -\frac{\sigma_v^2\left(\exp(-2\alpha_v T) - 1\right)}{2\alpha_v} \\ Q_{44} &= -\frac{\sigma_h^2\left(\exp(-2\alpha_h T) - 1\right)}{2\alpha_h} \end{aligned}$$

and

$$\begin{aligned} a &= \hat{n}_v + \hat{u}_v \\ b &= \hat{n}_h + \hat{u}_h \\ c &= 3 + \exp\left(-2\alpha_v T\right) - 4\exp\left(-\alpha_v T\right) - 2\alpha_v T \\ d &= 3 + \exp\left(-2\alpha_h T\right) - 4\exp\left(-\alpha_h T\right) - 2\alpha_h T \\ e &= \exp\left(-2\alpha_v T\right)\left(\exp\left(\alpha_v T\right) - 1\right)^2 \\ f &= \exp\left(-2\alpha_h T\right)\left(\exp\left(\alpha_h T\right) - 1\right)^2. \end{aligned}$$

4.3.5.2 Linearized Discretization

Recall that linearized discretization is the alternative approach in which the system is first discretized and then linearized.

The nonlinear state transition function, without noise, can be discretized via integration by:

$$s[t+T] = s[t] + \int_t^{t+T} f(s(\tau);\, \tilde{u}_v(\tau),\, \tilde{u}_h(\tau))\, d\tau. \tag{4.45}$$

Expanding the nonlinear state transition function, for the system defined in (4.39), yields the following discrete-time state update equations:

$$x[t+T] = x[t] + \int_t^{t+T} (\tilde{u}_v(\tau) + n_v(\tau)))\cos\left(\tilde{u}_h(\tau) + n_h(\tau)\right) d\tau \tag{4.46}$$

$$y[t+T] = y\left[t\right] + \int_t^{t+T} (\tilde{u}_v(\tau) + n_v(\tau))\sin\left(\tilde{u}_h(\tau) + n_h(\tau)\right) d\tau \tag{4.47}$$

$$n_v[t+T] = n_v[t] + \int_t^{t+T} -\alpha_v n_v(\tau) d\tau \tag{4.48}$$

$$n_h[t+T] = n_h[t] + \int_t^{t+T} -\alpha_h n_h(\tau) d\tau. \tag{4.49}$$

However, unlike the system in section 4.3.4.2, we cannot solve the integrals in equations (4.46)-(4.49) directly as written; they will require some manipulation.

We begin by noting that the noise dynamics are in fact linear. This means that we can bypass the integrals in equations (4.48) and (4.49) altogether and appeal to linear system theory to derive the update equations for n_v and n_h. Those update equations are found to be:

$$\begin{aligned} n_v[t+T] &= \exp\left(-\alpha_v T\right) n_v[t] \\ n_h[t+T] &= \exp\left(-\alpha_h T\right) n_h[t]. \end{aligned}$$

Using these results, we can expand the terms $n_v(\tau)$ and $n_h(\tau)$ in equations (4.46) and (4.47) to reflect their dependencies on the current state estimates. We can also replace the control input terms $\tilde{u}_v(\tau)$ and $\tilde{u}_h(\tau)$ with their current values by assuming that they remain constant over the sampling interval. The positional state update equations then become:

$$x[t+T] = x[t] + \int_0^T \left(\tilde{u}_v[t] + \exp\left(-\alpha_v \tau\right) n_v[t]\right) \cos\left(\tilde{u}_h[t] + \exp\left(-\alpha_h \tau\right) n_h[t]\right) d\tau$$

$$y[t+T] = y[t] + \int_0^T \left(\tilde{u}_v[t] + \exp\left(-\alpha_v \tau\right) n_v[t]\right) \sin\left(\tilde{u}_h[t] + \exp\left(-\alpha_h \tau\right) n_h[t]\right) d\tau.$$

Unfortunately, there is no simple closed form solution to either of these integrals because of the exponential that appears inside of the cosine and sine functions. However, it is possible to derive an approximation to the integrand that will yield a closed form solution.

Without loss of generality, let us consider the x-position update equation. The goal is to replace $\cos\left(\tilde{u}_h[t] + \exp\left(-\alpha_h \tau\right) n_h[t]\right)$ with a suitable approximation.

First, we can use the angle sum identity to factor the cosine above into:

$$\cos\left(\tilde{u}_h[t]\right) \cos\left(\exp\left(-\alpha_h \tau\right) n_h[t]\right) - \sin\left(\tilde{u}_h[t]\right) \sin\left(\exp\left(-\alpha_h \tau\right) n_h[t]\right). \tag{4.50}$$

The terms $\cos\left(\tilde{u}_h[t]\right)$ and $\sin\left(\tilde{u}_h[t]\right)$ are just constant factors and need no approximation. The remaining terms are the cosine and sine of the zero-mean exponentially correlated heading noise given by: $\exp\left(-\alpha_h \beta\right) n_h[t]$.

The heading noise takes its maximum value, $n_h[t]$, when $\tau = 0$ and decays toward zero as τ increases. With respect to the discretization integral, $n_h[t]$ is a constant, however, in terms of the overall system the noise state is an exponentially correlated random variable whose limiting distribution is a zero-mean Gaussian. This random variable models the disturbances that cause the target heading vary around $\tilde{u}_h[t]$. Its variance, and hence the largest expected angular deviation, is set by the constant model parameters α_h and σ_h.

For our tracking application, it is reasonable to assume that the largest possible range $n_h[t]$ will need to cover is $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ radians ($[-90, 90]$ degrees). Any greater uncertainty is probably best addressed by considering additional models. Typically, we will be interested in heading noise covering half that range or less.

Since possible the values of $n_h[t]$ are concentrated around zero and, with high probability, extend over a limited range, we can approximate $\cos\left(\exp\left(-\alpha_h \tau\right) n_h[t]\right)$ and

$\sin\left(\exp\left(-\alpha_h\tau\right)n_h[t]\right)$ using their series definitions. Those are given by:

$$\cos\left(\phi\right) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!}\phi^{2k}$$
$$\sin\left(\phi\right) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!}\phi^{2k+1}.$$

The first four terms, $k = 0, \ldots, 3$, provides acceptable accuracy for angles in the range $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ radians. Adding one or two more terms will extend the approximation to cover angles in the range $[-\pi, \pi]$ radians.

Letting $\phi = \exp\left(-\alpha_h\tau\right)n_h[t]$ and replacing the heading noise terms in (4.50) with their series approximations gives us a tractable integral.

In the y-position update equation, $\sin\left(\tilde{u}_h[t] + \exp\left(-\alpha_h\tau\right)n_h[t]\right)$ can be similarly approximated to yield a tractable integral. The sine is factored, using the angle sum identity, into:

$$\sin\left(\tilde{u}_h[t]\right)\cos\left(\exp\left(-\alpha_h\tau\right)n_h[t]\right) + \cos\left(\tilde{u}_h[t]\right)\sin\left(\exp\left(-\alpha_h\tau\right)n_h[t]\right). \tag{4.51}$$

The result is the following positional state update equations:

$$\begin{aligned}
x[t+T] =& x[t] + \cos\left(\tilde{u}_h[t]\right)\sum_{k=0}^{n}\left[\frac{(-1)^k}{2(2k)!}\frac{2n_v[t]\left(a-bd\right)}{\alpha_v + 2\alpha_h k} + \frac{\tilde{u}_v[t]\left(a-d\right)}{\alpha_h k}\right] \\
&- \sin\left(\tilde{u}_h[t]\right)\sum_{k=0}^{n}\left[\frac{(-1)^k}{\Gamma(2+2k)}\frac{n_h[t]n_v[t]\left(a-bcd\right)}{\alpha_h(1+2k)+\alpha_v} + \frac{n_h[t]\tilde{u}_v[t]\left(a-dc\right)}{\alpha_h(1+2k)}\right] \\
y[t+T] =& y[t] + \sin\left(\tilde{u}_h[t]\right)\sum_{k=0}^{n}\left[\frac{(-1)^k}{2(2k)!}\frac{2n_v[t]\left(a-bd\right)}{\alpha_v + 2\alpha_h k} + \frac{\tilde{u}_v[t]\left(a-d\right)}{\alpha_h k}\right] \\
&- \cos\left(\tilde{u}_h[t]\right)\sum_{k=0}^{n}\left[\frac{(-1)^k}{\Gamma(2+2k)}\frac{n_h[t]n_v[t]\left(a-bcd\right)}{\alpha_h(1+2k)+\alpha_v} + \frac{n_h[t]\tilde{u}_v[t]\left(a-dc\right)}{\alpha_h(1+2k)}\right],
\end{aligned}$$

where n determines the number of terms to use in the cosine/sine approximation, and the intermediate variables are given by:

$$\begin{aligned}
a &= n_h[t]^{2k} \\
b &= \exp\left(-\alpha_v T\right) \\
c &= \exp\left(-\alpha_h T\right) \\
d &= ac^{2k}.
\end{aligned}$$

Combining the individual state update equations into a vector-valued function, the discrete-time nonlinear system, still without noise, can be written as:

$$s[t+T] = g(s[t];\, \tilde{u}_v[t],\, \tilde{u}_h[t])\,.$$

As we saw in section 4.3.4.2, the discrete-time linear approximation has the form:

$$s[t+T] \approx F[t]s[t] + G[t]u[t] + w[t],$$

where $w[t] \sim N(0,\, Q[t])$. The components of the approximation, without noise, are described in (4.31).

For the system in (4.39) with current state estimate $\hat{s} = [\ \hat{x} \quad \hat{y} \quad \hat{n}_v \quad \hat{n}_h\ ]^\top$ and current control inputs $\hat{u}_v$ and $\hat{u}_h$ we get:

$$g'(\hat{s}) = F\,[t] = \begin{bmatrix} 1 & 0 & F_{13} & F_{14} \\ 0 & 1 & F_{23} & F_{24} \\ 0 & 0 & \exp(-\alpha_v T) & 0 \\ 0 & 0 & 0 & \exp(-\alpha_h T) \end{bmatrix}, \tag{4.52}$$

where

$$\begin{aligned} F_{13} &= \cos(\hat{u}_h)\, A - \sin(\hat{u}_h)\, B \\ F_{14} &= \cos(\hat{u}_h)\, C - \sin(\hat{u}_h)\, D \\ F_{23} &= \sin(\hat{u}_h)\, A - \cos(\hat{u}_h)\, B \\ F_{24} &= \sin(\hat{u}_h)\, C - \cos(\hat{u}_h)\, D \end{aligned}$$

and

$$
\begin{aligned}
A &= \sum_{k=0}^{n} \left[\frac{(-1)^k}{(2k)!} \frac{a - bd}{\alpha_v + 2\alpha_h k} \right] \\
B &= \sum_{k=0}^{n} \left[\frac{(-1)^k}{\Gamma(2+2k)} \frac{\hat{n}_h \left(a - bcd\right)}{\alpha_h(1+2k) + \alpha_v} \right] \\
C &= \sum_{k=0}^{n} \left[\frac{(-1)^k}{(2k)!} \frac{1}{\hat{n}_h} \frac{2k\hat{n}_v \left(a - bd\right)}{\alpha_v + 2\alpha_h k} + \frac{\hat{u}_v \left(a - d\right)}{\alpha_h} \right] \\
D &= \sum_{k=0}^{n} \left[\frac{(-1)^k}{\Gamma(2+2k)} \frac{bc - d\left(\alpha_h(1+2k)\hat{n}_v + b^{-1}\left(\alpha_h(1+2k) + \alpha_v\right)\hat{u}_v\right)}{\alpha_h\left(\alpha_h(1+2k) + \alpha_v\right)} \right. \\
&\quad \left. + \frac{(bc)^{-1} d\left(\hat{u}_v\alpha + \alpha_h(1+2k)\left(\hat{n}_v + \hat{u}_v\right)\right))}{\alpha_h\left(\alpha_h(1+2k) + \alpha_v\right)} \right].
\end{aligned}
$$

When the UT is used to linearize the system around the current state estimate, the components of the approximation are given by (4.32) and (4.33).

In either case, the discrete-time process noise covariance, $Q[t]$, can be computed using any of the approaches outlined in section 4.3.4.2. However, recall that the linearity of the noise dynamics allowed us to derive those update equations using techniques from linear systems theory. We can again appeal to linear systems theory to derive discrete-time covariance of the speed and heading noise states which are found to be:

$$
\begin{aligned}
\mathrm{Cov}\left(n_v\left[t\right]\right) &= \frac{\sigma_v^2}{2\alpha_v}\left(1 - \exp\left(-2\alpha_v T\right)\right) \\
\mathrm{Cov}\left(n_h\left[t\right]\right) &= \frac{\sigma_h^2}{2\alpha_h}\left(1 - \exp\left(-2\alpha_h T\right)\right).
\end{aligned}
$$

The two noise variables are independent so their covariances, plus the zero cross terms, give us four of the sixteen elements of $Q\left[t\right]$. These values can be used to verify the corresponding elements of the approximated covariance matrix or to replace those elements with their exact values.

4.3.5.3 Control Inputs

When used in a multiple model setting, we need the ability to force the system to a specific speed, by applying a deterministic control input, following a model switch. This ensures that the estimated speed at the end of the sample period is consistent with the assumptions used to determine the model parameters. As noted in our discussion of the previous system, it might also be desirable to incorporate the notion of the preferred

target heading. This would require the ability to force the system to a specific heading by applying a deterministic control input.

For the system defined in (4.39), support for this functionality is defined as follows.

Let $\tilde{u}_{n_v}[t]$ be the speed noise control input which is applied during the sample interval $[t,\, t+T]$. It represents the desired speed noise at time $t+T$, or in other words, the desired difference between the estimated speed and $\tilde{u}_v[t]$. Similarly, let $\tilde{u}_{n_h}[t]$ be the heading noise control input that represents the desired heading noise at time time $t+T$, or equivalently, the desired difference between the estimated heading and $\tilde{u}_h[t]$. The tilde notation is meant to signify that these are explicit control values intended to modify the state in a prescribed manner as opposed to the implicit control vector, $G[t]u[t]$, that represents the bias around the linearization.

For the Taylor series expansions, the linearization point is no longer the current state estimate at time t but rather the current position with the commanded speed noise and heading noise values. Thus, $\hat{s} = \begin{bmatrix} x[t] & y[t] & \tilde{u}_{n_v}[t] & \tilde{u}_{n_h}[t] \end{bmatrix}^\top$. The implicit control vector, $G[t]u[t]$, is computed as described above. However, it must be then be offset to account for the impact of the explicit control. The offset is given by:

$$\begin{bmatrix} -F_{13}\left(n_v[t] - \tilde{u}_{n_v}[t]\right) - F_{14}\left(n_h[t] - \tilde{u}_{n_h}[t]\right) \\ -F_{23}\left(n_v[t] - \tilde{u}_{n_v}[t]\right) - F_{24}\left(n_h[t] - \tilde{u}_{n_h}[t]\right) \\ -F_{33}\left(n_v[t] - \tilde{u}_{n_v}[t]\right) - F_{34}\left(n_h[t] - \tilde{u}_{n_h}[t]\right) \\ -F_{43}\left(n_v[t] - \tilde{u}_{n_v}[t]\right) - F_{44}\left(n_h[t] - \tilde{u}_{n_h}[t]\right) \end{bmatrix},$$

where F_{ij} denotes an element in $F[t]$.

When the UT is used to generate the approximation, the changes to the speed and heading values can be absorbed into $g(\cdot)$ by simply setting the speed noise and heading noise states to their commanded values prior to evaluating the original function.

4.3.6 Path Constraints

In the absence of any motion constraints, a target is typically assumed to hold its current heading during the given sample period. Thus, its estimated position at time $t+T$ is derived using dead reckoning based on its position at time t and its assumed speed over the sample period. The dynamic models introduced earlier in this chapter derive their estimates in exactly this manner. Recognizing that highly maneuverable

targets observed on a coarse time scale may indeed change course dramatically between samples, we looked at how to alter those models to reflect the assumption that the target changes its heading at the onset of each sample period to point toward the next measurement. The idea being that when the estimated position at time $t + T$ derived from dead reckoning fails to align with the measurement, it is due to an unmodeled change in heading that is corrected for by applying the appropriate control input. This treats all such changes in heading as equally likely, which in essence reduces the dynamic model to a 1-D speed model on a coordinate system, defined such that it passes through each measurement, embedded in the 2-D measurement space.

Both of these assumptions are variations of the same theme: the target moving along a straight-line trajectory. For short sample periods, this has proven to be reasonable approximation to the target's true motion in many cases. However, as the sample period gets longer, the approximation tends to break down because it fails to account for the influences from the terrain and the target's intent. Although a robust treatment of each of these effects will have to wait for future work, we can begin to model the influence of terrain by implementing a nonlinear dynamic model with a path following bias. This is based on the notion that a path represents a likely trajectory given the local terrain. In some cases it may represent the only feasible trajectory through the terrain. Note that the term "path" is used generically in the sense that it could refer to a trail, a roadway, or anything in between.

A path following bias is a form of constraint that allows us to make meaningful estimates of a target's state on a coarse time scale. It enables us to concentrate the probability of the target's estimated position based on our beliefs, albeit crude ones, about how the target will interact with the terrain. If we know the preferred target speed over different path segments, we can also concentrate the probability of the target's estimated speed based on that information and improve our estimates of the distance covered during the sample period.

The use of such constraints with ground targets is not new. However, ground targets have been almost universally assumed to be vehicles since the ability to detect dismounts at long range with radar is a recent phenomenon. When tracking vehicles, road constraints are commonly used to improve both the accuracy of the state estimates and data association performance [105, 31]. In some cases, these constraints are based not just on the physical roadway but the speed limits and traffic rules, such as one-way

streets [136]. The influence of the road network has been incorporated into the dynamic model in a variety of ways including: aligning the process noise covariance with the road direction [103], introducing the road constraints in the form of pseudo-measurements [52], and using a road-based coordinate system [11, 105]. The last approach is the most common and is the tightest constraint as it assumes that the target moves directly on the road centerline. When off-road movement must also be accommodated, the usual approach is to use unconstrained motion models in addition to road-constrained models. The set of active models may be adjusted dynamically based on whether the target is deemed to be moving on-road or off-road [176].

None of the existing approaches to road-constrained motion models are quite satisfactory for our purposes. We require a constraint that assumes little beyond the influence of the path on the shape of postulated trajectory. We want to avoid strong assumptions such as the target moves along the center of the path. Ideally, we would like to model movement of the target parallel to the path, when the target is close enough to be regarded as under its influence, as well as movement of the target directly along the centerline. This can be accomplished as follows.

Let the state space consist of Cartesian position, speed, and heading angle. The path-constrained dynamic model is nonlinear and implemented using the UT. For practical purposes, we use Wiener process models for the speed and heading. This significantly reduces the computational burden as it allows us to estimate the discrete-time process noise covariance using the UT and the nonlinear state update function evaluated for a single sample period based on (4.38). With the appropriate bookkeeping, the cost of the noise discretization can be minimized by exploiting the overlap with the state update (i.e. prediction).

We begin by assuming that the state at time t is such that the target can be regarded as under the influence of the path. Furthermore, we assume that the path and direction of travel are known. In practice, none of these assumptions are self-evident and must be established prior to invoking the model. We will address these issues after outlining how the state update function is implemented.

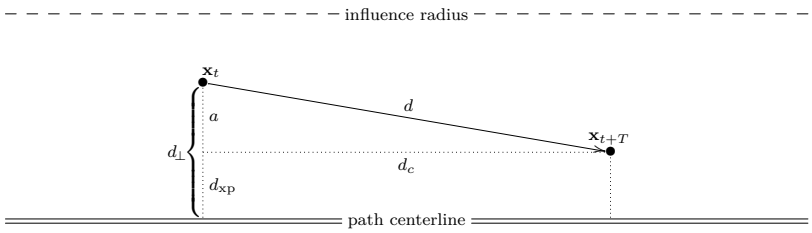
Given a vector representing the state of the target at time t, $g(\cdot)$ predicts the state at time $t+T$, without the influence of noise. In all cases, the prediction is influenced by the path but is performed in one of two ways depending on the relationship of the input state to the measurement at time $t+T$. The two different modes of prediction are

referred to as a "long" projection and a "short" projection. The distinction is based on whether or not the expected distance covered by the target during the sample interval is sufficient to reach the measurement or not. When projected along the path, if distance is greater than what is required to reach the next measurement, then it is considered a long projection. Otherwise, it is considered a short projection. The reason for treating these two cases differently will become clear as we explain how each one works.

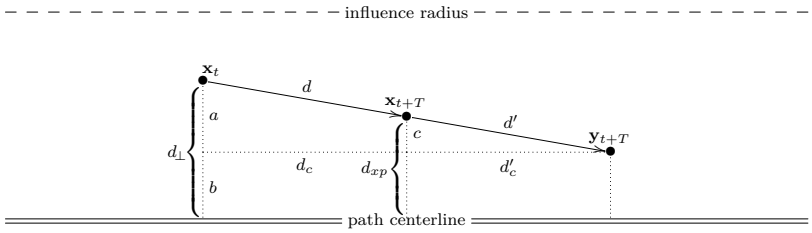
Let us start by describing the long projection. Figure 4.10a shows the geometry on which the projection is based. The variable $\mathbf{x}_t$ denotes the position of the target at time t and $\mathbf{x}_{t+T}$ denotes its position at the end of the sample period. These positions represent points in a 2-D space defined relative to the path. In this path-centric space, the abscissa describes the distance along the path and the ordinate describes the cross-path (i.e. perpendicular) distance. Imagine the space as constructed by straightening out the path to remove all its twists and turns. When the cross-path distance is zero, every point along the path can be mapped one-to-one to its corresponding position in the space space. For nonzero cross-path distances, a path-centric point is mapped to a position in the state space by first locating its position along the path and then projecting the cross-path distance, in the appropriate direction, along the normal vector at that location. Depending on the shape of the path, multiple path-centric coordinates may map to the same position in the state space, which does not present a problem.

Working in this path-centric space allows us to think in terms of the familiar straight-line trajectory and simplify the projection calculations. However, lengths are not preserved between the two spaces. For example, consider a trajectory in the path-centric space with a constant nonzero cross-path distance. When the path curves in the direction of this parallel trajectory, then its length in the position state space will be shorter. When the path curves in the other direction, then the length of the trajectory in the position state space will be longer. Hopefully, the two effects will tend to cancel out for actual paths. In practice, this appears to be a reasonable assumption.

To clarify what issues the path-centric space is meant to address, let us think about how we might project a point for a given distance in a given direction along the path. The point and projection are assumed to be in the position state space. If the initial point lies directly on the path and we wish to project along the centerline, then the problem is an easy one because the trajectory is well-defined and readily computable. When the initial point lies off the path, things get more complicated. We start by



(a) Long projection; the distance traveled is greater than what is required to reach the next measurement.



(b) Short projection; the distance traveled is insufficient to reach the next measurement.

Figure 4.10: Geometry of the path-constrained prediction.

associating the point to be projected with some reference point on the path. Usually, this will be the nearest point on the path. When the nearest point is not unique, then some application-specific logic must be used to select one to serve as the reference. Now that we can relate the point to the path, we could project it parallel to the path. In this case, the trajectory is well-defined but problematic to compute because it requires a representation of the parallel path which is not easy to derive. Things become even more complicated when the projection begins at one cross-path distance and ends at another, possibly on the other side of the path. In this case, the trajectory is no longer well defined or easy to calculate because we do not know how fast to change the cross-path distance with respect to its length. Using the path-centric space, all these calculations are straightforward. The distances may not be exact when mapped back into the state space, however, the computational advantages are substantial and distance errors have thus far proven to be tolerable in practice.

Apart from the matter of the space in which the projection is calculated, we still have to specify exactly how to project a given point along the path. The open question is how to define the trajectory and the long projection approach represents one answer. Referring back to figure 4.10a, we can see that the long projection begins at one cross-path distance and ends at another, which may be on the other side of the path. The initial cross-path distance, $d_\perp$, is assumed to be known as it is the

distance between the initial target position and the reference point on the path. The final cross-path distance, which is denoted by d_{xp}, is chosen so that it is consistent with the measurement at time $t+T$. This is based on the notion that the measured position has the correct cross-path distance. Being a long projection, we know that the final point is expected to lie further along the path than the measurement. The extent of the mismatch between the estimated and measured positions is assumed to be the result of mismodeling the speed of the target. Figure 4.11a shows an example of what a long projection looks like. Notice how the measurement dictates the how far off the path, in terms of perpendicular distance, the target's is predicted to go, while its speed dictates how far along the path it is predict to go.

Using the UT, we can estimate the mean and covariance of the cross-path distance from the measured position at time $t+T$ to the path. This defines a Gaussian distribution of d_{xp} which is then included as an input to the state update function in the long projection mode. Recall that the input to the state update function is $x \sim \mathcal{N}(\mu_x, \Sigma_x)$. The variable x is defined here as:

$$[\ x \quad y \quad v \quad d_{xp}\ ]^\top,$$

which is the Cartesian position, speed, and cross-path distance. The covariance matrix is assumed to be blockwise diagonal. Notice that the heading is not part of the input. This is because it does not affect the final state; the trajectory of the target is determined by the path and the projection algorithm rather than by the heading at time t. The heading may, however, be a factor in determining if the target can be regarded as under the influence of the path. The output variable, y, is the state of the target at $t+T$.

The position values produced by $g(\cdot)$ are based on mapping the projection point, specified relative to the reference point by d_c and d_{xp}, to the position state space. The cross-path distance, d_{xp}, is received as an input but the distance along the path, d_c, must be solved for. The projection distance, d, is computed by integrating the modeled speed over the sample period. This enables us to solve for the centerline distance as: $d_c = \sqrt{d^2 - (d_\perp - d_{xp})^2}$. The speed value produced by $g(\cdot)$ is based on the speed model. Finally, the heading value produced by $g(\cdot)$ is equal to the angle of a vector tangent to the path, where the tangent point is the point on the path nearest to the final projection point.

Let us now describe the short projection. Figure 4.10b shows the geometry

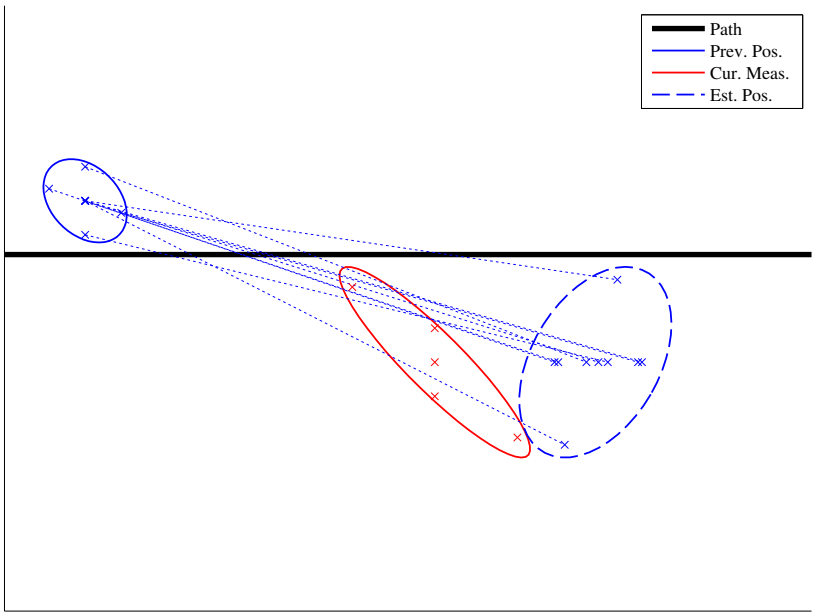
on which the projection is based. It is defined using the same path-centric coordinate space that we used for the long projection. In this mode, the trajectory is defined such that the target moves toward the measured position, $\mathbf{y}_{t+T}$. This is based on the notion that the measured position is ultimately where the target is headed. Being a short projection, we know that the final point is expected to lie somewhere along the path short of the measurement. The extent of the mismatch between the estimated and measured positions is assumed to be the result of mismodeling the speed of the target. As was the case with long projection, the projection begins at one cross-path distance and ends at another, which may be on the other side of the path. However, here it is the trajectory that determines the final cross-path distance. This is in contrast to the long projection in which the final cross-path distance was used to define the trajectory. Figure 4.11b shows an example of what a short projection looks like. The initial and measured positions are the same as those used in the long projection example; only the speed has been changed. Note that if we were to use the long projection approach with the lower speed, then the estimated position would have ended up in the lower-middle portion of the figure. This would imply a trajectory that cuts across the path and well below the measured position, which is not very plausible. Note also that if we were to use the short projection approach with a much higher speed, the trajectory would pass through the measured position and continue to diverge from the path. This would imply that despite the path constraint, the estimated position need not lie along the path!

The input to the state update function, x, is defined here as:

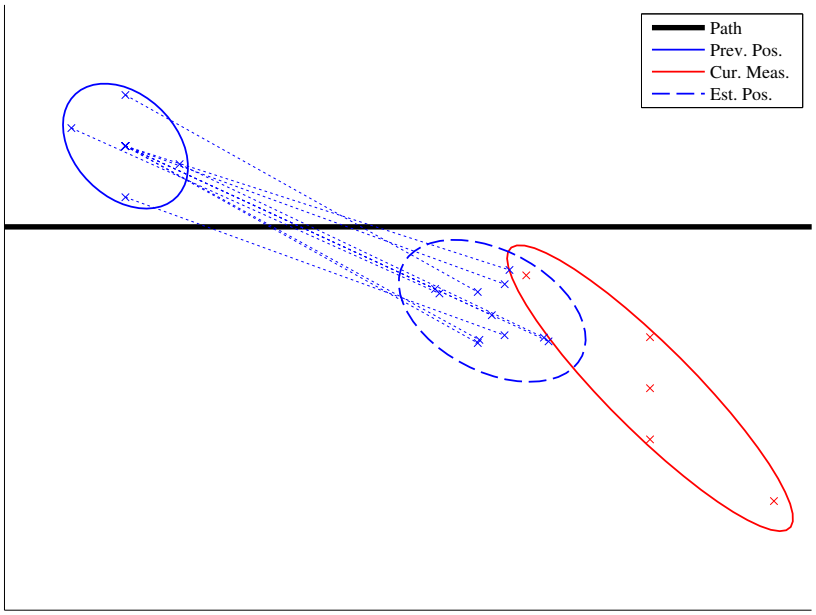
$$[\ x \quad y \quad v \quad x_m \quad y_m\ ]^\top,$$

which is the Cartesian position, speed, and the measured position (in Cartesian coordinates). The covariance matrix is assumed to be blockwise diagonal. Notice that the heading is again excluded from the input because it does not affect the final state. The output variable, y, is the state of the target at $t + T$.

The position values produced by $g(\cdot)$ are based on mapping the projection point, specified relative to the reference point by d_c and d_{xp}, to the position state space. Both the distance along the path and the cross-path distance must be solved for. The initial cross-path distance, $d_\perp$, is assumed to be known as it is the distance between the initial target position and the reference point on the path. The measured position



(a) Long projection; the distance traveled is greater than what is required to reach the next measurement.



(b) Short projection; the distance traveled is insufficient to reach the next measurement.

Figure 4.11: Example of the path-constrained prediction.

input is used to derive b by computing the distance between the measurement and the nearest point along the path, which is determined in the same manner as the reference point. The projection distance, d, is computed by integrating the modeled speed over the sample period. The remaining distance, d', is derived by subtracting d from the distance between $\mathbf{x}_t$ and $\mathbf{y}_{t+T}$, which are both inputs. The values for d_c and d_{xp} can then be solved for using the equations:

$$\begin{aligned}
\theta &= \cos^{-1}\left(\frac{d_\perp - b}{d + d'}\right) \\
d'_c &= d' \sin(\theta) \\
c &= \sqrt{d'^2 - d'^2_c} \\
d_c &= \sqrt{(d + d')^2 - (d_\perp - b)^2} - d'_c \\
d_{xp} &= b + c.
\end{aligned}$$

At this point we have seen how the path-constrained dynamic model is implemented. Now, we must address the matters of when such a model is appropriate and how to determine the path and direction needed by the projection algorithms.

The path-constrained model is appropriate when both the estimated position of the target and the next measured position are close enough, in some sense, to a common path. This notion of closeness is based on the shortest distance to some point on the path. For each path, the centerline is stored as a polygonal chain. Associated with each segment in the chain is a width of sorts. This width does not necessarily represent the physical width of the path, meaning the extent of the paved or packed surface, but rather its influence. And the width need not be symmetric with respect to the centerline. All points within some radius of the segment are deemed to be under the influence of the path. Since the target's estimated position and the measured position are both Gaussian random variables, decisions about whether these positions are under the influence of the path must take this uncertainty into account. In practice, we use a small deterministic set of sample points, such as the sigma points, and test each one to see if falls within the radius of influence of any path segment. If any points satisfy this criteria, then the position could plausibly be considered under the influence of the path.

Given a network of paths, there may very well be more than one common path between the estimated and measured positions. In this case, the true state of the

target is multimodal with one mode corresponding to the position projected along each path. We have chosen to handle this situation by using a path-constrained model for each possible path. The model selection algorithm is then responsible for generating a Gaussian approximation for each estimate, which may be based on choosing the most likely mode or a mixture of all modes, depending on the algorithm. Some algorithms look ahead an additional measurement, or more, to try to improve the selection by incorporating more information. For relatively sparse measurements and sparse path network, this works well enough. As the density of the path network increases, the computational requirements will increase and this approach is likely to break down. We will assume that the complexity of the path network is manageable and leave it to future work to develop algorithms designed to specifically address the challenges associated with predicting the movement of targets through complex networks.

Even after we have determined that a path-constrained dynamic model is appropriate and have identified the path, we still have to determine the direction of travel in order to make the projection. Usually, this will be unambiguous and simply fall out of the process of identifying the path. When this is not the case, it is usually because the target has not moved very far and abandoning the path-constrained model in favor of the stopped target model may be appropriate. We can also opt to extend the set of motion models by including on one for each combination of path and direction.

Chapter 5

Experimental Evaluation

5.1 Introduction

At this point, we have introduced the associative tracking algorithm along with several dynamic models intended for our motivating application: wide-area surveillance of human targets using radar. Now we will evaluate the performance of the algorithm and the dynamic models in the context of that application.

The evaluation will be conducted using a combination of simulated and real radar data. However, we will rely on the former in order to establish a clean baseline to which future experimental studies can be compared. By focusing on simulated measurements, we can isolate our analysis of the tracking algorithm from issues related to the performance of the detector, which is outside of the scope of this dissertation. It also enables us to avoid problems related to not knowing for certain what is really in the data set. This is a major concern when working with wide-area surveillance data because cost, or practicality, often precludes us from knowing exactly what is happening within any realistic region of interest. When the tracker identifies movement in some area outside of the control of the experiment, we are left in the uncomfortable position of not being able to determine if the output was erroneous or not. We can also never be certain if any unscripted targets were missed. This means that no matter how accurately we record the state of the scripted targets, our ground truth is not guaranteed to be complete or correct. In addition, we generally will not be able to associate individual measurements with the targets that they originated from because of the uncertainties in the detection process.

Simulated measurements ensure that complete, accurate, and precise ground truth is available. Of course, their realism can be called into question. And some tasks, such as evaluating the dynamic models, require measurements of actual targets since the models themselves cannot be used. To address these issues, we will simulate only the sensor and not the targets themselves. High-precision target trajectories will be recorded using a differential global positioning system (DGPS) from which measurements of the target can be simulated. Thus, we only rely on simulation of the measurement process, which we assume is well-understood and can be accurately modeled. The underlying target motion is all real.

In addition to the measurement uncertainty, the sensor is also characterized by its false alarm rate and probability of detection. These too fall within the scope of the

simulation and are assumed to be well-understood and amenable to accurate modeling. For the false alarm rate, it is quite reasonable to assume that we have a realistic model even when the sensor does not actually achieve the ideal of a constant rate. This is because the false alarm rate can be established experimentally for a given sensor and operating environment. For the probability of detection, on the other hand, this is really a simplifying assumption intended to give us a performance baseline because accurate probability of detection models are difficult to obtain. In principle, the probability of detection could also be established experimentally, but in practice this is only feasible for a tiny fraction of the surveillance region.

5.2 Simulated Measurements

The simulator is based on a simple model of a generic radar that directly measures the position of a target in range and azimuth, relative to the radar location. Optionally, the simulator supports the direct measurement of the target's radial velocity as well, but we will not consider that further in order to match the capabilities of a low-cost non-coherent radar. Measurement of the elevation angle is not supported as all targets are assumed to be located on the earth's surface. In keeping with the notion of modeling only the most widely available capabilities, measurement of target features, such as the radar cross section or range profile, is not included. The signal-to-noise ratio (SNR) is likewise excluded from the model because it is often unavailable to the tracking algorithm.

The radar model itself is composed of three parts: a false alarm model, probability of detection model, and a measurement model.

False alarms are modeled in terms of their rate of arrival and spatial distribution. The rate of arrival is expressed by a discrete count distribution that describes the number of false alarms per scan. This distribution is not restricted to any parametric form, but the default is a Poisson distribution. The default spatial distribution of false alarm measurements is a uniform distribution defined over the sensor footprint, however, this too may take any convenient form. In practice, we expect both the false alarm count and spatial distributions to be defined based on estimates derived from experiments conducted in the field.

Given a range and azimuth, the probability of detection model returns a scalar

value indicating the probability of detecting a target present at the given location. It will typically be implemented using a lookup table and interpolation. The default, however, is simply a constant P_D. How to obtain the data for a spatially varying model is beyond the scope of this dissertation and we recognize that in many instances it will not be available. In some ways, the general form of the probability of detection model is a place holder for future developments in radar signal modeling. At the present time, the tools and techniques, such as 3-D radar ray tracing simulators [77], needed to estimate the probability of detection over the surveillance region are not widespread and still in their infancy. However, given detailed terrain data, line of sight visibility modeling is presently feasible, which can be used to create spatially varying binary P_D models (i.e. one value for terrain-masked regions and another for visible regions). This is a step in the right direction. When a principled model is not available and we wish to avoid the assumption of a constant P_D, we will consider randomly generated binary P_D models.

The measurement model is parameterized by the sensor location, scan rate, resolution, accuracy, and range limits. It is the resolution, stated in terms of beam width and range resolution, that determines the radar's ability to resolve closely spaced targets. To be resolved by the radar, a target must be separated from the other targets by at least the beam width in azimuth or the range resolution in range. Unresolved targets will be merged by the simulator and replaced by a contact whose position is the centroid of the merged measurements. Measurements are assumed to be corrupted by independent zero-mean Gaussian noise in range and azimuth. Thus, the accuracy is specified by the standard deviation for each coordinate. Their values are chosen such that they reflect all error sources impacting the measurement including SNR, propagation conditions, interference, or internal noise [39].

Together, the parts of the model work as follows. In each scan, the sweep of the beam is simulated and any targets that it intersects, within the specified range limits, are identified as contacts. Once the target contacts have been established, the number of false alarm measurements is drawn from the count distribution. For each one, a location is drawn from the spatial distribution to form a false alarm contact. The set of all contacts, both target and false alarm, is then clustered and unresolved contacts are merged. Note the implication that the measured position of a target may be affected, independent of the noise, by either false alarms or other target measurements that

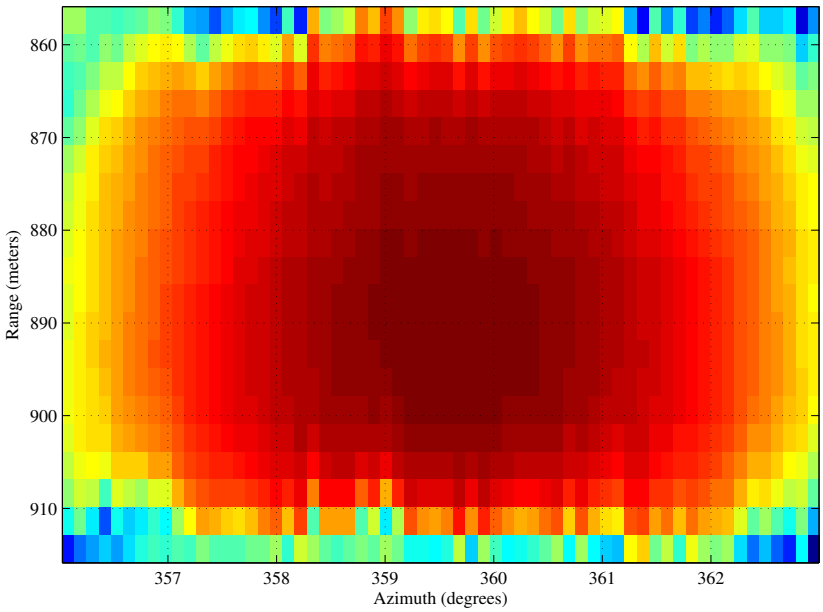


Figure 5.1: Point target response from a surveillance radar.

appear within the same resolution cell. The probability of detection is then evaluated at the position of each of the remaining target contacts. A random draw with that probability of success determines whether or not the contact is reported. If the contact is reported, then its position is corrupted by adding a sample from a truncated version of the noise distribution. The truncation ensures that the error is appropriately bounded. Note that false alarm contacts that remain after clustering are not affected by the probability of detection model nor are they affected by the measurement noise.

For the simulations used in this chapter, the relevant model parameters were chosen to be representative of some of the commercial surveillance radars that we have worked with. Those being a beam width of 1.1° and a range resolution of 20 m. The accuracy parameters were $\sigma_R = 2.5\,\text{m}$ and $\sigma_A = 0.138°$. Truncation was performed at $\pm 3\sigma$. Figure 5.1 shows a point target response from one of the surveillance radars that we studied. We can see that it is consistent with these parameters and a Gaussian uncertainty model appears to be reasonable.

The accuracy of the DGPS data is discussed in section 4.2.1. The data was recorded at 5 Hz.

5.3 Dynamic Models

To evaluate the consistency and accuracy of the dynamic models introduced in chapter 4, we used them within a Kalman filter framework. In particular, we analyzed the state estimation error, which is defined as:

$$\tilde{s}[t] = s[t] - \hat{s}[t],$$

where $s[t]$ is the true state and $\hat{s}[t]$ is the estimated state of the target at time t.

For a linear-Gaussian dynamic system, consistency implies that a statistical characterization of the true state estimate error agrees with the output of the filter up to second order [6]. Thus, a consistent model will produce errors that are zero-mean (i.e. unbiased) and have a mean squared error (MSE) that matches the filter-calculated covariance. Consistent models will also yield a time series of innovations that are zero-mean and white. At each time step, the innovation is the difference between the actual measurement and its predicted value, computed using the measurement model and the current state estimate. Evaluating consistency is important because it is equivalent to evaluating the optimality of the models since the filter-calculated covariances affect the evolution of the state estimates over time.

Of course, the dynamic model is not the only design decision that impacts the consistency of the estimates. The choice of the measurement model does as well. However, it usually receives little attention because the relationship between the state and the measurement variables is generally straightforward. We mention it here because the relationship is nonlinear in our application, which does require some design decisions be made in order to keep within the Kalman filter framework. Since the radar position measurements are made in a local polar coordinate system while the position state is defined in a Cartesian coordinate, conversion between the two is necessary. We chose to handle the conversion implicitly using a nonlinear measurement model in conjunction with the unscented transform (UT). This provided a consistent debiased Gaussian approximation to the Cartesian equivalent of the measurement distribution [94]. Maintaining the consistency of the resulting statistics was a particular concern here because Lerro [113] demonstrated that there is a risk of the transformations becoming inconsistent when the standard deviation of the azimuth is less than one degree, which is the case here.

The accuracy of the models is characterized in terms of the average errors in the estimated position and speed of the target. We also looked at the average divergence between the estimated and true state distributions. Errors in the estimated heading were not explicitly considered because we do not have precise enough ground truth for this component of the state, even for simulated measurements. That does not present a problem because heading errors are really only important to the extent that they impact the estimated position. The same could also be said for errors in the speed estimates, however, speed models form the basis of our dynamic models so we want to know how well they work. Evaluating the accuracy is important because it establishes the performance limits of state estimation in the absence of any data association ambiguity.

In order to compute the errors required for these evaluations, the true state of the target needs to be known. Therefore, we used simulated measurements. For each set of DGPS data, we generated 50 sets of simulated measurements in order to apply statistical tests for consistency and to compute the average errors. The simulations were based on a given sensor position and sample period. Since radars have a large discrepancy between their in-range and cross-range uncertainties, we wanted the data to include measurements made using a variety of orientations of the sensor with respect to the target's motion. This was accomplished by specifying the sensor position in terms of a range and orientation that are defined relative to the centroid of the DGPS-recorded positions. We then took the range to be a free parameter which is associated with a fixed set of orientations: 0°, 45°, 90°, and 135°. Every given combination of range and sample period thus specifies four different sensor configurations and 50 sets of simulated measurements were generated based on each one.

Each model configuration under consideration was evaluated for a given set of DGPS data, sensor range, and sample period. This constitutes an experiment. The DGPS data was selected to represent one of the three terrain types: flat (T1), rolling (T2), and mountain (T3). The sensor range and sample period were selected to represent typical values for wide-area surveillance applications. For the range, those values are: 3 km, 5 km, and 7 km. For the sample period, they are: 15 s, 30 s, and 60 s. Associated with each experiment are 200 sets of simulated measurements; 50 corresponding to each of the four sensor configurations. This gave us a sample of 200 independent state estimation errors for each time step (i.e. scan).

A model configuration refers to both the specific type of dynamic model used

to describe the motion of the target and its parameters. Since targets are expected to move in either the "walk" or "stopped" regimes, each configuration specifies a pair of models that are used together within the Kalman filter framework. The generalized pseudo-Bayesian (GPB) estimator of first order was used to produce a single estimate based on both models. It was chosen because it outperformed the interacting multiple model (IMM) estimator in a preliminary investigation of multiple model algorithms using our data.

Note that in the discussions to follow in the next section, when the "walk" model uses different parameter values for each terrain, they are listed using set notation where the elements correspond to flat, rolling, and mountain terrain, respectively.

In these experiments, the sensor position was treated as an extraneous variable. For the accuracy evaluation, we controlled for it by simply averaging the metrics over all sensor positions. For the consistency evaluation, we held it constant by testing each sensor configuration independently. The reason for treating the sensor position differently between the two evaluations will be explained when we come to the variance test.

Let us now describe the statistical tests used to verify the conditions required to establish consistency. For simplicity, we used classical hypothesis tests rather than their Bayesian counterparts. All statistical tests were conducted using $\alpha = 0.05$.

With respect to consistency, our primary interest lies in testing the zero-mean and variance matching conditions. Both conditions can be verified simultaneously using the normalized (state) estimation error squared (NEES) [6], which is defined as:

$$\epsilon[t] = \tilde{s}[t]^\top P[t]^{-1} \tilde{s}[t],$$

where $P[t]$ is the covariance of the state estimate. When the dynamic model is consistent, $\epsilon[t]$ follows a chi-squared distribution with n_s degrees of freedom, where n_s is the dimension of the state vector. To determine whether or not this is the case, we compute the sample mean $\bar{\epsilon}[t]$. Under the hypothesis that the model is consistent, $N\bar{\epsilon}[t]$ will follow a chi-squared distribution with Nn_s degrees of freedom. For the two-sided test, the hypothesis is rejected at the significance level α if $N\bar{\epsilon}[t] \geq \chi^2_{1-\alpha/2}$ or $N\bar{\epsilon}[t] \leq \chi^2_{\alpha/2}$, where χ^2_i is the critical value at i for a chi-squared distribution with Nn_s degrees of freedom.

Unfortunately, using the NEES to assess the consistency of a model turns out

to be highly problematic in practice. We mention it here because it is quite literally the textbook approach. However, it should be avoided. Consolidated statistics, such as the NEES, have a tendency to obscure problems affecting the individual state variables. In other words, the value of the statistic may support the hypotheses that the model is consistent, while individual tests of a given condition applied to a given state variable say otherwise. Using the well-known CWNA dynamic model, we will show an example of this which illustrates the danger of setting the noise parameters to achieve the desired NEES value.

In practice, we did not expect any of these models to be perfectly consistent because we did not expect the underlying linear-Gaussian assumption to always be satisfied. Yet, we wanted to make a principled assessment of how close they came to that ideal. With that goal in mind, we tested the zero-mean and variance matching conditions, individually, for each state variable. This provided us with a fine-grained assessment. Note that the covariance between different state variables was not tested.

A one-sample t-test of the population mean, given an unknown population variance, was used to determine if the errors are zero-mean. The test statistic, which follows a t-distribution, is:

$$t = \frac{m}{s/\sqrt{N}},$$

where m is the sample mean, s is an unbiased estimate of the sample standard deviation, and N is the sample size. For the two-sided test, the hypothesis that the population mean is equal to zero is rejected at the significance level α if $|t| \geq t_{1-\alpha/2}$, where t_i is the critical value at i for a t-distribution with $N-1$ degrees of freedom. For the one-sided test that the population mean is less than zero, the hypothesis is rejected if $t < t_\alpha$. For the one-sided test that the population mean is greater than zero, the hypothesis is rejected if $t > t_\alpha$.

A chi-squared test of the population standard deviation was used to determined if the covariance matching condition is satisfied. The test statistic, which follows a chi-squared distribution with $N-1$ degrees of freedom, is:

$$\chi^2 = (N-1)\frac{v}{v_0},$$

where v is an unbiased estimate of the sample variance, N is the sample size, and v_0 is the nominal population variance. For the two-sided test, the hypothesis that the population variance is equal to v_0 is rejected at the significance level α if $\chi^2 \geq \chi^2_{1-\alpha/2}$ or

$\chi^2 \leq \chi^2_{\alpha/2}$, where χ^2_i is the critical value at i for a chi-squared distribution with $N-1$ degrees of freedom. For the one-sided test that the population variance is less than v_0, the hypothesis is rejected at the significance level α if $\chi^2 \geq \chi^2_{1-\alpha}$. For the one-sided test that the population variance is greater than v_0, the hypothesis is rejected at the significance level α if $\chi^2 \leq \chi^2_{\alpha}$.

An obvious question is: where did we get the nominal population variance, v_0? For a given sample, we have N errors from which to estimate v. However, we also have N filter-calculated variances. To get a single value with which to test against we took v_0 to be the average variance over the sample. This approach is reasonable only when the sample is drawn from a single sensor position because the distribution of filter-calculated variances in this case is unimodal with low variance. Otherwise, a single variance would not be representative due to the influence of the sensor position. For this reason, we evaluated the consistency for each sensor configuration independently.

Note that the aforementioned statistical tests can exhibit sensitivity to non-Gaussian data and outliers. In order to increase our confidence that they were in fact working as expected, we compared the results from a subset of the tests to those obtained by using bootstrap variants of those tests [50]. The results were found to be in agreement. Because of the high computational cost and the large number of tests to be performed, bootstrap tests were only used for verification.

Having described the statistical tests, let us now look at how they were applied within each experiment.

At each time step, a given hypothesis was tested independently four times, one for each sensor configuration, using a sample size of 50 for each test. This resulted in a total of $4ns$ tests for every state variable, where ns is the number of scans. Note that for a given sensor configuration, the tests at each time step are dependent because of the Markov dependence between the states.

Clearly, we want to combine the results from all the tests of a given hypothesis to draw a single conclusion. Unfortunately, there is no universally accepted approach to dealing with the problem of multiple comparisons. A natural way forward in this case would be to use an extension to Fisher's method for combining p-values from multiple tests into a single statistic; the original method is not applicable because we have a mix of independent and dependent tests. We tried Kost's method [107] for combining dependent p-values and a similar method from Shaoyu et al. [114]. The results were not

satisfactory for our purposes.

The problem was that we almost always ended up rejecting the null hypothesis and with p-values that leave no room for interpretation. For example, after 1244 tests of the zero-mean condition for one of the state variables, just under 13% were rejected at the chosen significance level yet the combined hypothesis was rejected with a p-value of zero (because of the finite numerical precision). So, despite the fact that just a small minority of the individual tests lead to rejections, the combined outcome was indistinguishable from the most extreme deviation from the hypothesis possible. Such a result is hardly a useful characterization of the overall performance.

As we mentioned earlier, we did not expect any of these models to be perfectly consistent. The methods of Kost and Shaoyu et al. are too strict as they specifically test the hypothesis that the means in all of the tests were exactly zero (or the variances matched exactly). That sets the bar much too high. What we really wanted was to get a sense of how close the model came to the ideal of consistency. A simple way to do that is to look at the fraction of the individual tests that were rejected. This gave us a sense of how each model performed with respect to the condition being tested. When 50% of the tests are rejected, we considered the evidence to be inconclusive. As the fraction of rejections goes to zero, we considered the evidence increasingly supportive of the notion that the tested condition was satisfied.

Another matter related to multiple comparisons that needed to be addressed is that of false rejections. When considering a large number of statistical tests, some p-values will be below the chosen significance level purely by chance. Thus, by looking at the set of tests of a given hypothesis as a whole, we might be overstating the number of rejections[1]. Various methods have been developed to correct for this tendency. We used the Holm–Bonferroni method [85] since it is a standard approach and requires no assumptions. Note that for computing the two-sided p-values for the chi-squared distribution, we used the conditional p-value to deal with the asymmetry [108].

In addition to testing the zero-mean and variance matching conditions, the whiteness of the innovations was also tested by comparing the autocorrelation function (ACF) to the confidence bounds for the significance level α. In every experiment, the ACF was estimated independently for each sensor configuration. The ACF was

[1]This is above and beyond the inherent bias of classical hypothesis testing in favor of overstating the evidence against the null [167].

estimated using Welch's method with a 2 min window and 50% overlap, with the periodograms averaged over all 50 realizations. The fraction of outliers plus a visual inspection of the ACF was used to assess the performance for a given experiment.

The results of the consistency evaluation for each experiment are summarized in a table that contains: the state estimation errors, the standard deviation matching errors (i.e. the difference between the square roots of the MSE and filter-calculated variance for the given state element), the fraction of zero-mean tests that were rejected, and the fraction of variance matching tests that were rejected. The reported errors are averages computed using all of the available data. They are included to provide a sense of scale for the bias and variance mismatch. Note that because of the way that the error was defined, negative numbers indicate filter-calculated values that are larger than the true values. The results from the whiteness tests were not included in the table to reduce information overload and because they are of secondary importance given the fact that we have access to the true state. Typically, the whiteness of the innovations is afforded primary importance because that is the only consistency-related test that can be performed directly on real data. The bias and variance matching are what really matter and are thus our focus. We will remark on the whiteness results and include figures to illustrate the ACF when appropriate.

The results of the accuracy evaluation for each experiment are summarized in a table that contains: the position error, the speed error, the divergence, the percent change in the position error, the area of the position uncertainty, and the percent change in area. In addition to reporting the average value of each metric computed using all of the available data, there is column that includes the standard deviation of each metric with respect to the different sensor positions. The reported position error is the average Euclidean distance between the estimated and true positions. The reported speed error is the mean absolute error. The divergence is the average Jensen-Shannon divergence of a single estimate. These three metrics quantify the accuracy of the model while the remaining metrics are included to help understand the positional accuracy in more detail. The percent change in the position error represents the relative change between the measurement and estimate errors. To visualize the uncertainty in the position estimates, the area is computed from the covariance using the ellipse at $\pm 2\sigma$. Finally, the percent change in the area represents the relative change between the area covered by the measurement uncertainty and that covered by the estimated position.

All of the result tables are collected in appendix D.

5.3.1 CWNA

To establish a performance baseline, we evaluated the Wiener process speed model from section 4.3.2 extended into two dimensions. It is more commonly known as the continuous white noise acceleration (CWNA) model and serves as the dynamic model for ground targets in the vast majority of tracking algorithms in the literature. In addition to the CWNA model in its default form, we also evaluated the system rotation technique proposed in section 4.3.3 to handle highly maneuverable targets. It provides an alternative to the usual approach of inflating the process noise to accommodate target maneuvers. Unlike the high-noise maneuver models, this approach alters the mean of the estimate by assuming that the target travels toward the next measurement. Finally, we evaluated a high-noise variation of the default CWNA with a process noise selected based on the NEES statistic. Although this configuration was not meant to represent a maneuver model, it does illustrate some of the potential problems with high-noise models in addition to the problems associated with using consolidated consistency statistics.

The state consists of x-position, y-position, x-velocity, and y-velocity. Since the evaluation requires an estimate of the target's speed, which is not directly represented here, prior to analysis all estimates were converted into a position-speed-heading format using a high-order (i.e. large number of sigma points for increased accuracy) unscented transform.

The configurations evaluated were defined, with respect to the "walk" regime, as follows:

- C1: Default model with $\sqrt{T}\sigma_w = \{0.25, 0.25, 0.29\}$ applied in both x and y.
- C2: Maneuver modeling via system rotation using the same noise parameters as C1.
- C3: Default model with $\sqrt{T}\sigma_w = 5.0$. This configuration was only evaluated for terrain T1.

A stop model, which predicts no change in position, was used for the "stopped" regime with $\sqrt{T}\sigma_w = 0.1$. It was implemented using the CWNA/C1 model with a modified position prediction.

Tables C.1–C.3 contain the consistency summary statistics and tables C.4–C.6 contain the performance summary statistics.

With respect to the position estimates for targets moving in flat terrain, the CWNA/C1 model was found to be modestly biased (approximately 0.14 m, on average) with well-matched variances. The innovations were acceptable as white with few outliers; none falling significantly outside of the confidence bounds. The speed estimates, however, were clearly biased with the filter-calculated speeds exceeding the true speed by approximately 0.16 m/s, on average. The standard deviation of the speed estimate overstated the true uncertainty by about the same amount. In rolling and mountain terrain, the bias in the position estimates increased modestly while remaining essentially unchanged for the speed estimates. The position variances remained well-matched and the speed variances continued to be overstated by the model. The innovations were less white but not significantly so. Overall, the model improved the expected position error, compared to using the measured position directly, in all but one experiment. Depending on the sensor configuration, the improvements ranged from approximately 1% to almost 18%, with the biggest gains associated with targets observed at near range and with a short sample period. Interestingly, the position error did not vary significantly across the different terrain types. The speed errors, on the other hand, increased slightly as the terrain became more rugged.

Compared to the previous configuration, the CWNA/C2 model showed moderate overall improvements in the accuracy and consistency of the position estimates. The quality of the speed estimates, however, was essentially unchanged. Generally speaking, it tended to produce estimates that were closer to the true state, as evidenced by the lower divergence metric in most of the experiments. Although the expected position error was ultimately comparable between the two models, the system rotation led to an almost three percentage point improvement, on average, in the change in the position error. That is to say, it yielded a greater improvement relative to the measurement error than the default model. It also managed to improve the expected position error in all experiments. With respect to the bias in the position estimates, the fraction of rejections dropped from 0.31 to 0.13 on average . The variances remained well-matched and the whiteness of the innovations improved. In the rolling and mountain terrain, the improvement in the quality of the position estimates was a bit more pronounced. This is presumably because a constant heading is a less accurate prediction in more rugged

terrain.

The third and final configuration is identical to the first except for the noise level used to define the "walk" regime model. In this case, the noise level was selected such that most of the NEES statistics fell within the confidence bounds, which required a four-hundred-fold increase in the variance. Figure 5.2 shows a subset of the NEES values obtained from the two models in one of the experiments conducted in flat terrain. Clearly, the CWNA/C3 model is superior when evaluated using this metric. It even managed to rein in the outliers that correspond to overestimates of the speed for slow movement that fell between the "walk" and "stopped" regimes. However, as table C.1 indicates, the bias of the position estimates has increased ever so slightly but has worsened considerably for the speed estimates faring no better than $-1.85\,\mathrm{m/s}$. It appears that the only real improvement brought on by the increased process noise was in the variance matching in the position estimates. However, it was already excellent with no more than 5% of the individual tests rejected. What is even more damning are the performance statistics in table C.4. In all but two experiments, the expected position error is now worse than if we had simply used the measurements directly! Where it did manage to reduce the expected position error, the reduction was no more than 1.4%. On average, the estimated target speed was off by more than the maximum speed of the target. By rolling all aspects of consistency into a single statistic, these problems were masked and led us to a rather poor estimator. This not only illustrates why it is important that we always evaluate each aspect of consistency separately but also that care must be taken when increasing the process noise variance beyond what is justified by the target dynamics.

5.3.2 PSH

Given the focus on speed in our characterization of the target dynamics, the model based on a polar representation of velocity described in section 4.3.4 is a natural choice. It defines the state as x-position, y-position, speed, and heading, hence the label "PSH". This enables us to characterize the target's rate and direction of travel separately, which is consistent with our thinking about its dynamics. Within this framework, we evaluated Wiener process models for both the speed and heading. We also evaluated the effectiveness of applying a control input to align the predicted heading with the next measurement which reflects the same notion of a preferred target heading that motivated

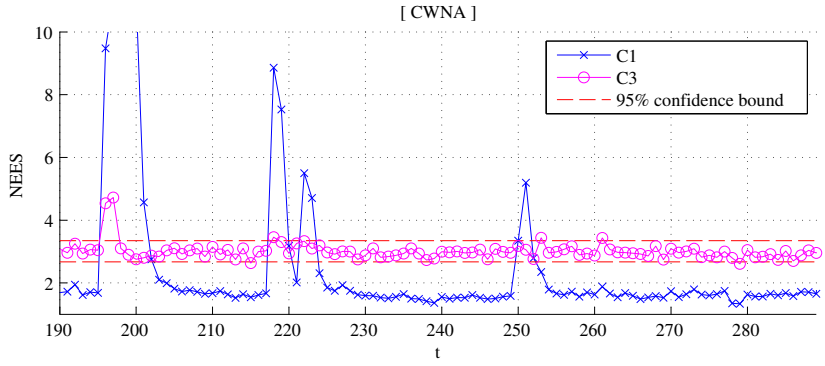


Figure 5.2: Example of the normalized (state) estimation error squared (NEES) statistics for the default system, CWNA/C1, compared to the high noise variant, CWNA/C3.

the maneuver modeling via system rotation that we considered for the CWNA model. Finally, we evaluated the stationary Gaussian alternative to the Wiener process speed model discussed in section 4.3.2.4. This approach uses a control input to force the predicted speed at each time step to the mean speed for the "walk" regime.

Note that the PSH model used in these experiments was linearized using the unscented transform (UT). The discrete-time process noise was also derived using the UT as explained in section 4.3.4.2. We will consider the effect of a different linearization algorithm next when look at a version of the PSH model with exponentially correlated noise.

The configurations evaluated were defined, with respect to the "walk" regime, as follows:

- C1: Wiener process speed and heading models with $\sqrt{T}\sigma_v = \{0.25, 0.25, 0.29\}$ and $\sqrt{T}\sigma_h = 10$ (degrees). Note that all configurations shared these noise parameters.
- C2: Wiener process speed and the heading aligned with the next measurement.
- C3: Stationary speed with $\tilde{u}_v[t] = \{1.44, 1.38, 1.32\}$ and Wiener process heading.
- C4: Stationary speed (with the same speed control input as C3) and the heading aligned with the next measurement.

A stop model, which predicts no change in position, was used for the "stopped" regime with $\sqrt{T}\sigma_v = 0.1$ and $\sqrt{T}\sigma_h = 90$ (degrees). It was implemented using the PSH/C3 model with $\tilde{u}_v[t] = 0$.

Tables C.7–C.9 contain the consistency summary statistics and tables C.10–C.12 contain the performance summary statistics.

We begin by comparing the first two configurations to see how the different heading models performed in conjunction with the Wiener process speed. It turns out that the Wiener process heading in the PSH/C1 model fared somewhat poorly. Although the expected position error was only about 1 m higher, on average, the model failed to improve it relative to the measurement error in most of the experiments. The PSH/C2 model, on the other hand, improved the position error in all of the experiments. The improvements were comparable to what we saw for the CWNA/C2 model, ranging from approximately 1% to 16%, with the biggest gains associated targets observed at near range and with a short sample period. With respect to the position estimates, the PSH/C1 model was modestly biased (approximately 0.2 m, on average) with well-matched variances. Using the preferred target heading reduced the bias to just under 0.05 m, on average, and reduced the fraction of variance matching tests that were rejected. Neither model had innovations that could be accepted as white; only approximately so with large numbers of slight outliers for small lags. The PSH/C1 model underestimated the target's speed by approximately 0.17 m/s, on average, with largest errors occurring in uneven terrain. It overstated the standard deviation of those estimates by 0.38 m/s, on average. The PSH/C2 model produced more accurate and less biased speed estimates but overstated the uncertainty by a similar amount. Overall, the PSH/C2 model produced estimates that were much closer to the true state as evidenced by the significantly lower divergence metrics.

Looking at the how the different heading models performed in conjunction with the stationary speed model, the winner is clear: align the heading with the next measurement. For the PSH/C3 model, the expected position errors were on the order of ten times higher than those from PSH/C4 with a bias of almost 17 m, on average. The variance of the estimates was poorly matched, understating the true error standard deviation by almost that amount. The speed estimates were almost 0.6 m/s too low, on average. Based on these results, the PSH/C3 model should be eliminated from any serious consideration. By comparison, the PSH/C4 model performed well. Its accuracy, in particular, was excellent. With regards to consistency, the fraction of zero-mean and variance matching tests that were rejected suggests that the states estimates are somewhat biased with only moderately well-matched variances. In absolute terms, the

bias in the position estimates was just under 0.1 m, on average, and the bias in the speed estimates was just under 0.05 m/s (too low), on average. Thus, it appears that many of the zero-mean test rejections are a consequence of the strictness of the point null hypothesis. The standard deviation of the position estimates were understated by 0.94 m, on average, while it was overstated for the speed estimates by 0.19 m/s on average. These are arguably only moderate in their agreement as fraction of rejections suggested. In just one-third of the experiments, the innovations could be regarded as white. Otherwise, there were numerous minor outliers; mostly for small lags.

Comparing all four configurations, the PSH/C4 model was the most accurate. It had the lowest expected position error and the lowest absolute speed error. In terms of the position error, it bested the next best model, PSH/C2, by about 1 m, on average. More importantly, the PSH/C4 model yielded the biggest improvement in the position errors relative to the measurement errors by a large margin upping the gains to between 6% to 27%. This seems to suggest that the stationary speed yields some tangible benefits. The fact that the two best performing configurations aligned the heading with the next measurement, provides more evidence that this practice is indeed beneficial. Regarding the consistency of the estimates, the results were mixed. The PSH/C2 model produced position estimates that were closer to the ideal of consistency, yet the PSH/C4 model produced more consistent speed estimates. In terms of the divergence metric, PSH/C2 model was generally superior.

5.3.3 PSH-ExpCorr

Our analysis of the DGPS data from human walkers revealed that both the speed and heading have a similar non-negligible autocorrelation structure. Despite the fact that the correlation time (on the order of 30 s) is relatively short compared to the typical sample period, it makes sense to evaluate the effectiveness of incorporating that structure into the dynamic model. Thus, we considered the position-speed-heading model with exponential correlation described in section 4.3.5. The speed and heading are modeled there as Uhlenbeck-Ornstein processes which drift toward their respective equilibrium levels. For the speed model, this is simply the expected speed in the given motion regime. For the heading model, however, the equilibrium level is best determined uniquely for each prediction based on the preferred target heading. Once again, we assumed that this preferred heading was aligned with the next measurement. Since

changing the equilibrium level for each prediction is equivalent to switching between multiple models at each time step, control inputs were used to ensure that the predicted heading was consistent with the assumptions as explained in section 4.3.5.3. There are two ways that this could be done and we evaluated both of them. The first, which we took to be the default, is to apply an input that drives the heading noise state to zero. The second is to apply an input to adjust the heading noise state to its equivalent value given the current equilibrium. This means that as the sample period grows, the predicted heading goes from the current heading to the preferred value. Finally, the PSH-ExpCorr model was used to compare the effectiveness of two different approaches to obtaining a discrete-time linear approximation. To do this, we evaluated the default model based on discretized linearization and then again based on linearized discretization with the UT.

The configurations evaluated were defined, with respect to the "walk" regime, as follows:

- C1: Inputs used to drive n_h to zero. Model derived using the discretized linearization approach.
- C2: Same as C1 but with the model derived using the linearized discretization.
- C3: Inputs used to adjust n_h to its equivalent value. Model derived using linearized discretization.

All of the above were based on the parameters: $\sqrt{T}\sigma_v = \{0.25, 0.25, 0.29\}$, $\sqrt{T}\sigma_h = 10$ (degrees), $\tilde{u}_v[t] = \{1.44, 1.38, 1.32\}$, and $\alpha_v = \alpha_h = 1/15$.

A stop model, which predicts no change in position, was used for the "stopped" regime with $\sqrt{T}\sigma_v = 0.1$ and $\sqrt{T}\sigma_h = 90$ (degrees). It was implemented using the PSH/C3 model with $\tilde{u}_v[t] = 0$ since the exponential correlation is irrelevant for stopped targets.

Tables C.13–C.15 contain the consistency summary statistics and tables C.16–C.18 contain the performance summary statistics.

Looking first at the approximation schemes, we found that there were no significant performance differences between the two approaches. Both the consistency and accuracy were comparable across all of the experiments. However, the PSH-ExpCorr/C2 model did enjoy a very slight advantage over the PSH-ExpCorr/C1 model when it came

to the expected errors and the divergence metric. Of course, the PSH-ExpCorr/C2 model maintains a significant advantage with respect to the ease and simplicity of the implementation.

Applying an input to adjust the current n_h to its equivalent value given the current heading equilibrium proved to be ineffective. In terms of both consistency and accuracy, the PSH-ExpCorr/C3 was inferior to the other configurations. The difference in accuracy, however, was not large with the expected position error being 0.72 m higher, on average, and the expected speed error being 0.017 m/s higher, on average. More significantly, the position estimates resulted in considerably less improvement over the measurement error. Overall, the estimates produced by the PSH-ExpCorr/C3 model were further from the true state as evidenced by the significantly higher divergence metrics.

Based on the fraction of zero-mean and variance matching tests that were rejected, the PSH-ExpCorr/C1 and PSH-ExpCorr/C2 models were somewhat biased with only moderately well-matched variances. However, as was the case for the PSH/C4 model, in absolute terms the bias was quite modest, being just under 0.1 m, on average, for the position estimates and just under 0.05 m/s (too low), on average, for the speed estimates. The standard deviation of the position estimates was understated by about 0.55 m, on average, while it while it was overstated for the speed estimates by 0.078 m/s, on average. The innovations were reasonably white with mostly just minor outliers.

Compared to their uncorrelated counterpart, the PSH/C4 model, the exponentially correlated models offered no compelling advantage. The PSH-ExpCorr/C1 and PSH-ExpCorr/C2 models had better variance matching with similar bias. Although they tended to exhibit lower divergence from the true state distributions, their accuracy was slightly lower than that of the PSH/C4 model with an additional 0.29 m position error, on average, and an additional 0.011 m/s speed error, on average.

5.3.4 PSH-Path

A path represents a likely trajectory given the local terrain. Thus, path constraints provide a simple means of incorporating some of the influence of terrain into the dynamic model. This in turn should enable us to make better predictions than equivalent unconstrained models, especially over long time intervals, because the assumed target trajectory is arguably more realistic. Since the motivating application

requires dynamic models that work well when the targets are observed on a coarse time scale, we evaluated the path-constrained dynamic model described in section 4.3.6. As described, the model was implemented as a nonlinear extension of the position-speed-heading model with Wiener process speed and heading. The primary motivation for this decision was that it simplifies the implementation of the process noise model. Since the path constraints largely dictate the predicted state, the heading model has no influence beyond its impact on the noise model. The choice of speed model, however, is rather consequential as it determines how far along the path the target is assumed to have traveled. So, in addition to the Wiener process speed model, we also evaluated the stationary Gaussian alternative discussed in section 4.3.2.4. Because the path constraints themselves were intended for predictions over long time intervals and the correlation time is relatively short compared to those intervals, a speed model with exponential correlation was not considered.

For each set of DGPS data, the paths used by the target were manually digitized with the aid of publicly available satellite imagery. Any ambiguity with respect to the paths connecting the current state to the next measurement was handled by adding a separate model for each possible route. The radius of influence was set generously to 25 m so that all estimates would make use of the path constraints. No alternative unconstrained models were used for these experiments.

The configurations evaluated were defined, with respect to the "walk" regime, as follows:

- C1: Wiener process speed and heading models with $\sqrt{T}\sigma_v = \{0.25, 0.25, 0.29\}$ and $\sqrt{T}\sigma_h = 10$ (degrees). Note that both configurations shared these noise parameters.

- C2: Stationary speed with $\tilde{u}_v[t] = \{1.44, 1.38, 1.32\}$ and Wiener process heading.

A stop model, which predicts no change in position, was used for the "stopped" regime with $\sqrt{T}\sigma_v = 0.1$ and $\sqrt{T}\sigma_h = 90$ (degrees). It was implemented using the PSH-Path/C2 model with $\tilde{u}_v[t] = 0$.

Tables C.19–C.21 contain the consistency summary statistics and tables C.22–C.24 contain the performance summary statistics.

Using the Wiener process speed model in conjunction with path constraints resulted in estimates that, based on the fraction of hypothesis tests rejected, were some-

what biased with only moderately well-matched variances. However, as we have seen before, in absolute terms the bias was quite modest, being 0.061 m, on average, for the position estimates and 0.072 m/s (too low), on average, for speed estimates. The standard deviation of the position estimates was overstated by 1.35 m, on average, while it was overstated for the speed estimate by 0.28 m/s, on average. The innovations were acceptable as white with few outliers; none falling significantly outside of the confidence bounds. Compared to its unconstrained counterpart, the PSH/C2 model, the path-constrained model was similarly biased but overstated the variance to a larger degree. The PSH-Path/C1 model, however, was significantly more accurate, as expected. It improved the expected position error by 3.72 m, on average, and improved the expected speed error by almost 0.1 m/s, on average. Not surprisingly, the path constraints led to estimates that were much closer to the true state as evidenced by the lower divergence metrics across the board.

Switching to the stationary speed model improved the accuracy but generally worsened the consistency of the estimates. This is the same behavior that we observed with the unconstrained models when comparing PSH/C2 to PSH/C4. In the case of the PSH-Path/C2 model, the degradation in consistency was limited to the position estimates. There, the bias increased to 0.31 m, on average, while the impact on the variance matching is harder to summarize. In some experiments, the average mismatch actually dropped while in others, notably at long range and in the non-flat terrain, the variance was substantially underestimated. Compared to its unconstrained counterpart, the PSH/C4 model, the path-constrained model was a bit less consistent overall but generally more accurate. With respect to the expected position error, the PSH-Path/C2 model was clearly superior. However, it actually had slightly larger expected speed errors in most experiments. Based on the divergence metrics, the constrained model produced estimates that were much closer to the true state. It also produced the largest improvement over the measurement error with gains between about 15% and 54%.

5.4 Feasible Graph

As we saw in chapter 2, the feasible graph is integral to the inference process within the associative tracking algorithm. Recall that it plays two critical roles. First, the feasible graph determines the subset of the solution space to be searched

by identifying the regions of non-negligible probability. When done properly, this winnows the enormous space of possible associations and makes inference considerably more tractable. However, if the subset does not contain all of the true associations, then the complete and correct tracks can never be recovered. The second role is to facilitate the generation of proposals that enable the sampler to efficiently explore that subset of the solution space. When the feasible graph leads to poor proposals, the given processing budget may be insufficient for the sampler to visit all the high probability regions or the resulting sample may not accurately characterize the uncertainty. In fact, unless the proposals are good enough, in the sense that they facilitate efficient exploration of the space, then no reasonable amount of processing time will be sufficient.

The feasible graph is thus a key determinant of the accuracy and efficiency of the associative tracking algorithm as a whole. Its performance must therefore be evaluated with respect to these two roles. We did this, independent of the inference process, by analyzing a variety of feasible graphs constructed from representative data sets. Each feasible graph was constructed using a different tracklet model so that we could find out which models were most effective. We also looked at how the construction parameters that apply to all tracklet models, such as thresholds, impacted the performance.

With respect to the first role, it is the structure of the feasible graph that determines the subset of the solution space that will be searched. Specifically, it is the edges that define the feasible associations of measurements to targets since each track corresponds to a (directed) path in the graph. Our primary concern is that the set of feasible associations does in fact include the true associations. In terms of the graph structure, this means that for each target there is an edge connecting every pair of vertices corresponding to consecutive measurements of the target. These are known as degree-one edges, where the degree indicates the order of the difference in the sequence of successive target detections. Edges connecting measurements that did not originate from the same source are known as degree zero edges. Performance along this dimension was measured by looking at what fraction of edges of each degree were present in the feasible graph. We also looked at what fraction of the true track length is represented by the edges in the graph. This calculation was based on the longest path consisting of edges with degree ≥ 1 for the given target.

Assuming that our primary concern has been addressed, our next concern is that the set of feasible associations contains most of probability mass. Said another

way, we would like to know that the regions of the solution space that were excluded represent just the thinnest tails in the distribution of associations. Unfortunately, this cannot be verified in general because of the combinatorial size of solution space and we must rely on judicious application of the techniques designed to control the density of the graph. In practice, we have eschewed the use of thresholds applied to the tracklet probability in favor of gating based on speed and time. We did, however, limit the out-degree of each vertex by keeping only the n-best edges because the tractability of inference is an overriding concern. For the applications that we envision, it is preferable to identify the highest probability configurations and underestimate the uncertainty to some extent by failing to thoroughly explore the tails of the distribution rather than risk missing something important while trying to explore the full space with insufficient computational resources.

With respect to the second role of the feasible graph, both its structure and the weights of its edges impact the quality of the proposals and thus the efficiency of the sampler. When thresholds are applied to the tracklet probabilities during construction of the graph or only the n-best out-edges are kept for each vertex, the structure and edge weights are in fact related. With the latter being the method of choice for controlling the graph density, our evaluation focused on the edge weights. Our concern is that the weight of each edge, which by definition is the tracklet probability, is correlated with the joint likelihood of the corresponding measurements. Large edge weights should indicate a high likelihood that the edge connects two measurements that are part of the same track. When this assumption holds, the feasible graph will have a structure that is consistent with its two roles. The graph will contain the edges corresponding to the most probable associations and the weights can be used to inform the proposals by serving as a proxy for the likelihood.

Evaluating the performance of a feasible graph with respect to its second role is not so straightforward because, ultimately, success is determined by the quality of the proposals that the graph enables. However, trying to separate out the impact of the edge weights from all the other inference-related variables by analyzing the final sample introduces a host of additional problems. Instead, we analyzed the weights directly to determine how well the underlying tracklet model discriminated between true associations and erroneous associations. In other words, we assessed whether true associations were likely to have higher tracklet probabilities. This was done first, looking

only at the edge weights and the true value of the corresponding association event, using the H-measure [78] which is a technique to measure the performance of a binary classifier such as the tracklet model. Then, we looked at how well the discrimination compared to that of the likelihood model itself by computing the target/false alarm likelihood ratios, for a given track length, using both the likelihood model and the tracklet approximation.

The H-measure was selected to avoid difficulties associated with using a much more well-known summary statistic for binary classifier performance: the area under the receiver operating characteristic (ROC) curve. The ROC curve is a method of visualizing the performance of a binary classifier for different choices of the discrimination threshold. It is created by plotting the true positive rate against the false positive rate at various threshold settings. The ROC curve itself provides an effective means of evaluating the tradeoffs between recall and precision as a function of the threshold value. A common method of summarizing the overall performance is to compute the area under the curve (AUC). This represents the probability that a randomly chosen positive instance will be assigned a higher probability by the classifier than a randomly chosen negative instance [79], which is a very useful metric for our purposes. Unfortunately, the AUC has been shown to be an incoherent means of comparing classifiers because the implicit misclassification cost distribution used is dependent on the classifier [78]. The H-measure was developed to provide a coherent alternative which also returns a scalar value in [0, 1], where unity represents perfect separation between the classes.

These evaluations, of course, require that the origin of each measurement be known, so we once again used simulated measurements. A single set of DGPS data, which consisted of five walkers moving in mountain terrain for one-half hour, was used for all experiments. It was selected to represent a typical surveillance scenario. The sensor position was set approximately 5 km south of the activity and the walkers were sufficiently well-separated to be resolved independently. Their motion included move-stop-move behavior and backtracking. Simulated measurements were generated using a variety of sample periods, false alarm rates, and probability of detection models. Assuming a constant probability of detection with $P_D = 0.9$, nine sets of simulated measurements were generated using the sample periods $T = \{15, 30, 60\}$ seconds and false alarm rates $\{5, 15, 30\}$ false alarms per scan. Those false alarm rates are equivalent to 9.272×10^{-7}, 2.781×10^{-6}, and 5.563×10^{-6} false alarms per m^2 per scan, respectively. A more realistic spatially non-uniform P_D model was also considered. Since a principled

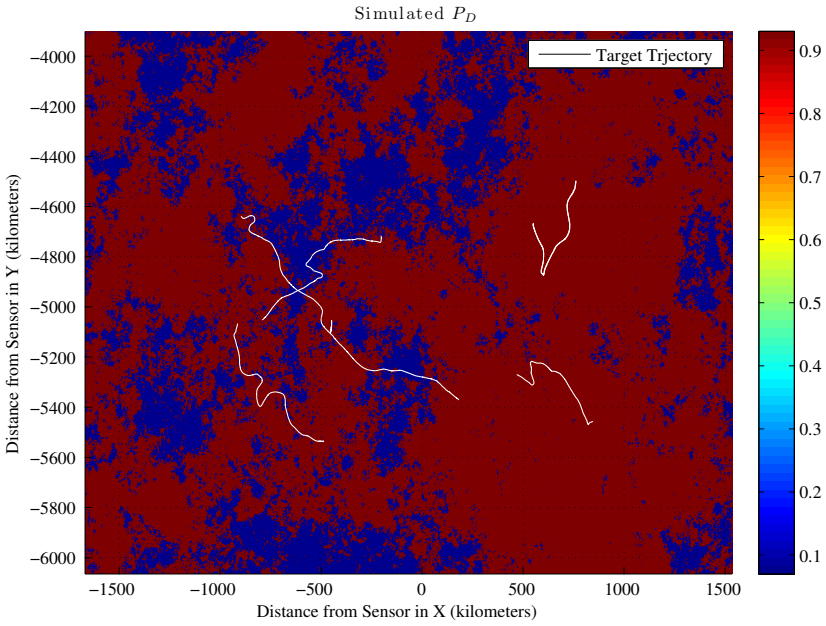


Figure 5.3: Random binary P_D model used for the feasible graph evaluation.

model was not available, the randomly generated binary model with $P_D = \{0.93, 0.07\}$ shown in figure 5.3 was used. Note that the low probability of detection regions cover approximately one quarter of the surveillance region. Nine sets of simulated measurements were generated using this binary P_D model with the same sample periods and false alarms rates that were used for the constant P_D model. This gave us a total of eighteen data sets.

5.4.1 Tracklet Models

A tracklet model describes the conditional probability that two given measurements originated from the same target. The appropriate form of this model is highly application specific. Depending on how much data is available, a tracklet model might be learned using a general-purpose algorithm for binary classification. It could also be explicitly constructed using domain knowledge. We used the latter approach and constructed various models based on target speed and the probability of detection. These are perhaps the simplest possible tracklet models and are intended to serve as a baseline to which future work with more sophisticated models, learned or explicitly constructed, can be compared.

We considered two classes of tracklet models which, for lack better terms, are referred to as "generative" and "discriminative". Recall that the tracklet model was said to be a discriminative model because it does not describe how the observed data is probabilistically generated from the unobserved association event but rather directly describes the dependence of the unobserved association event on the data. Of course, this does not preclude us from modeling the data generating process and then using Baye's rule to derive the requisite conditional probability for the tracklet model. The so-called generative tracklet models uses this approach. The so-called discriminative tracklet models, on the other hand, do not model the data generating process. Instead, they express the conditional probability of an association event using features extracted from the data.

Generative

The generative models studied here use the likelihood components of the associative tracking model (see section 2.3.1) to define the tracklet probability. We considered just a single formulation of the model but multiple instances; one for each dynamic used to describe the evolution of the latent states.

The tracklet probability was defined as:

$$p(A_{ij} \mid \tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j) = \frac{p(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j \mid A_{ij})\, p(A_{ij})}{p(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j \mid A_{ij})\, p(A_{ij}) + p(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j \mid \neg A_{ij})\, p(\neg A_{ij})}, \tag{5.1}$$

with

$$p(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j \mid \neg A_{ij}) = p(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j \mid E_1) + p(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j \mid E_2) + p(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j \mid E_3) + p(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j \mid E_4), \tag{5.2}$$

where E_1 denotes the event that the i-th and j-th measurements are both false alarms, E_2 denotes the event that the i-th measurement originated from a target and the j-th is a false alarm, E_3 is the same as E_2 but with the origins reversed, and E_4 denotes the event that both measurements originated from different targets.

The target measurement likelihood was used to define $p(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j \mid A_{ij})$. For each term in (5.2), the measurement likelihoods were assumed to be independent. The false alarm measurement likelihoods are given by the associative tracking model. The target measurement likelihoods for a single measurement were computed using one-point initialization of the dynamic model [6]. In all experiments, we used $p(A_{ij}) = 0.5$.

All twelve of the dynamic models that were evaluated in the previous section were evaluated here as generative tracklet models.

Discriminative

In order to keep the models in each class comparable for these experiments, we considered a single discriminative model based on just two features: speed and probability of detection. This is basically the same information that informs the generative tracklet models. The idea was to evaluate the discriminative alternative to the speed models described in section 4.3.2. A discriminative tracklet model with path constraints was not evaluated.

The first feature, speed($\mathbf{y}_i$, $\mathbf{y}_j$), is the expected speed of a target assuming that it followed a straight line trajectory between the two measured positions. Note that the speed is a function of the physical measurements, hence the missing tildes. This is because the probability of detection is being handled differently than the associative tracking model. The second feature is computed independently for each measurement to get $p_d(\mathbf{y}_i)$ and $p_d(\mathbf{y}_j)$. It is the expected probability of detection given the P_D model and the measurement uncertainty.

The tracklet probability model was defined as:

$$p(A_{ij} \mid \tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j) = p(D_i)\, p(D_j)\, p(S_{ij})\,, \tag{5.3}$$

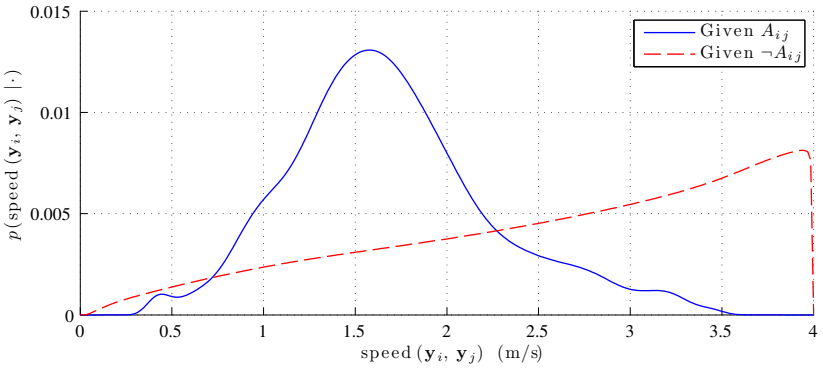
which D_i denotes the event that a target in the position given by i-th measurement was detected, D_j denotes the event that a target in the position given by j-th measurement was detected, and S_{ij} denotes the event that the speed implied by the two measurements is consistent with a target moving between them.

The probabilities $p(D_i)$ and $p(D_j)$ are simply given by $p_d(\mathbf{y}_i)$ and $p_d(\mathbf{y}_j)$, respectively. The probability $p(S_{ij})$ is given by:

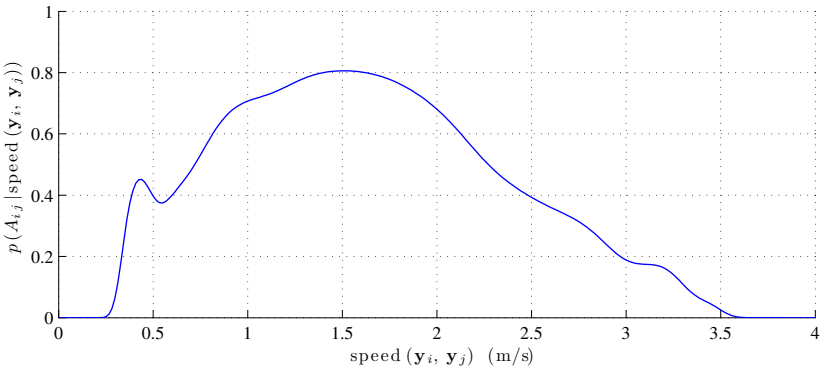
$$\begin{aligned} p(S_{ii}) &= p(A_{ij} \mid \text{speed}(\mathbf{y}_i, \mathbf{y}_j)) \\ &= \frac{p(\text{speed}(\mathbf{y}_i, \mathbf{y}_j) \mid A_{ij})\, p(A_{ij})}{p(\text{speed}(\mathbf{y}_i, \mathbf{y}_j) \mid A_{ij})\, p(A_{ij}) + p(\text{speed}(\mathbf{y}_i, \mathbf{y}_j) \mid \neg A_{ij})\, p(\neg A_{ij})}, \end{aligned} \tag{5.4}$$

where $p(\text{speed}(\mathbf{y}_i, \mathbf{y}_j) \mid A_{ij})$ is the distribution of the speed given that the measurements originated from the same target and $p(\text{speed}(\mathbf{y}_i, \mathbf{y}_j) \mid \neg A_{ij})$ is the distribution of the speed given that the measurements have different origins. As was the case for the generative model, the prior for the association event was $p(A_{ij}) = 0.5$ in all experiments.

Figure 5.4 shows an example of how $p(S_{ij})$ was defined for human walkers in mountain terrain observed using a scan period of 15 s and with 15 false alarms per



(a) Likelihood



(b) Posterior

Figure 5.4: Example of the probabilities used to define the S_{ij} event in the discriminative tracklet model.

scan. The likelihood distributions were estimated using training data that was generated based on DGPS data from walkers in that terrain. The DGPS used for the training was different than that used to generate the simulated measurements for the evaluations. In figure 5.4a, you will immediately notice that the distribution of the expected target speed as computed by the tracklet model is not the same as what we observed in our GPS study (see figure 4.1c). The mode is considerably higher and the distribution has a larger variance. This is an important revelation as it will influence the speed threshold for gating based on speed.

5.4.2 Coverage

The coverage of a feasible graph refers to the fraction of edges of each degree that are present. Degree-one edges are the most critical as their presence is required to ensure that the true measurement to target associations are among those that can be expressed by the graph. Degree-two edges connect every second measurement of a target and are important because they give us a chance to recover when a degree-one edge is omitted. As the degree increases, it becomes increasingly less likely that such edges will be present in the graph, although this depends on how aggressively we try to control the density of the graph. However, even when high degree edges are present, their weights tend to decrease as the edge degree increases because the uncertainty in the target's motion grows and the assumptions of the tracklet model break down. Thus, we focused the evaluation on edges of degree one and two as they have the most practical relevance. In order to quantify the impact of any missing edges between target measurements, we also looked at the fraction of the total track length (in terms of the number of measurements) expressible by the graph.

If the density of the graph were of no concern, then coverage would not be an issue. It would be sufficient to only exclude edges that violate hard constraints such as the scan constraint or maximum association time. Such edges represent pairwise associations with zero prior probability. However, in practice, the density of the graph is a major concern because it is directly related to the complexity of the inference. Thus, we sought ways to exclude additional edges while maintaining acceptable coverage.

To this end, we considered the impact of applying a maximum speed gate and limiting the maximum out-degree of each vertex. No minimum speed was considered because targets may come to a stop. Hard constraints–scan and maximum association

time–were also enforced. Note that although the latter only prevents the inclusion of edges representing zero probability associations, it can impact the coverage. Applying a threshold to the tracklet probability was another option available to reduce the density of the graph. This approach, however, was dismissed because it increases the chances of dismissing a degree-one edge. We deemed it preferable to keep even low probability edges emanating from a given source vertex until enough higher probability alternatives are found to displace them.

Based on the GPS study in chapter 4, 2.25 m/s is a reasonable upper bound on the walking speed of a human target. However, this turned out not to be a useful value for the speed gate because it led to the omission of approximately $1-5\%$ of the degree-one edges.

The speed was computed in exactly the same manner as the speed feature for the discriminative tracklet model described in the previous section. Despite a full accounting of the measurement uncertainty, the distribution of these estimates does not match the true target speed distribution as we saw already in figure 5.4a. The estimated speed is an overestimate. One potential remedy would be to consider the variance of the speed estimate when testing the gate condition. Another would be to simply set a higher threshold based on the distribution of the speed estimates. This is the approach that we took. A threshold of 4 m/s was used throughout these experiments, which was sufficient not to exclude any degree-one edges based on the speed gating.

Using a 3 min maximum association time and a maximum out-degree of 10, all eighteen data sets were evaluated using each of the tracklet models. For the data sets based on the constant P_D model, all of the feasible graphs contained 100% of the degree-one edges regardless of the tracklet model. In each track, at least 83% of the degree-two edges were present as well. In fact, only when $T = 60$ s were any degree-two edges omitted. For the data sets based on spatially non-uniform binary P_D model, all of the degree-one edges were covered for all configurations except those for which $T = 30$ s and 60 s. With the longer sample periods, at least 89% of the degree-one edges, in each track, were present.

It turns out that these gaps in coverage were due to too many missed detections. With a sample period of 30 s or 60 s, the detector can only miss the target in 6 or 3 consecutive scans, respectively, before the 3 min maximum association time is exceeded. In the more challenging detection environment of the non-uniform P_D model, this started

to become a problem with long sample periods as we might have expected. The fact that this was not a problem for any of the other configurations is purely by chance. When the probability of detection is not unity, there is always the possibility of failing to detect the target with sufficient frequency to maintain the track. These results do show, however, that in a realistic setting we can expect to get a feasible graph with good coverage. The fractions of the total track lengths for the incomplete tracks were: 6/10, 12/14, and 23/26. The corresponding fractions of degree-two edges were: 38%, 75%, and 88%, respectively.

The 3 min maximum association time was chosen based on the notion that beyond this point too much uncertainty has built up with respect to the target's motion to make a meaningful association. This value is indeed arbitrarily defined and a reasonable threshold will be application-specific. In these experiments, increasing it to 6 min would have led to all the degree-one edges being covered for all configurations. However, whether or not we would be justified in associating two measurements of a maneuverable target after a 6 min information blackout is debatable. An example of where we might wish to use such a generous maximum association time is for an application in which we want to monitor movement along a known trail and are willing to disregard off-trail motion. In this case, the path constraints reduce the uncertainty in the allowable motion of the target and the treatment of negative information in the associative tracking model will ensure that the likelihood reflects whether or not the missed detections are plausible with respect to the P_D model. The P_D model might also inform the choice of the maximum association time by allowing us to calculate the size of the potential drop-outs based the average speed of the target over hypothesized routes.

Limiting the out-degree of every vertex to 10 appears to reliably result in complete coverage of the degree-one edges. Even in our experience outside of these experiments, we found this to be true. In general, feasible graphs constructed using that parameter, in conjunction with the speed gate and maximum association time, will provide a significant reduction in the size of the solution space. However, their density still leaves a difficult inference problem. To get a sense of scale, let us look at the number of edges in the feasible graphs constructed from the data sets based on the non-uniform P_D model with $T = 15$ s. These half-hour data sets resulted in graphs with 6393, 19050, and 36141 edges for the cases of 5, 15, and 30 false alarms per scan, respectively.

The obvious question is, what happens to the coverage when we reduce the maximum out-degree of each vertex? This, of course, will depend on the quality of the edge weights, which will be examined in the next section. Ideally, the degree-one edges will have the largest weights and be retained. Those edges that are excluded will ideally remove associations with negligible likelihood from the solution space. We can only verify the first of these two conditions and for all but one of the tracklet models it does hold to a remarkable degree.

With the exception of the discriminative tracklet model, the degree-one edges were assigned the largest weights in all but a handful of cases. Limiting the out-degree of every vertex to 3 turned out to be sufficient maintain coverage in those experiments. That change reduced the edge counts in our earlier example to 875, 2061, and 3807, respectively, which is a substantial savings. In the case of discriminative tracklet model, the degree-one edges were less likely to be assigned the largest weight. Keeping just the top 3 resulted in a loss of approximately 5% of those edges. We would have had to keep the top 8 to maintain full coverage in these experiments. These results highlight the importance of knowing how well the chosen tracklet model performs before adjusting this parameter. Even the notion that 10 is a safe default should not be taken for granted. The results also show that given the right tracklet model, we can hope to use the maximum out-degree to safely reduce the complexity of the inference problem in realistic environments.

5.4.3 Edge Weights

A feasible graph was constructed for every combination of tracklet model and data set. Every such combination constitutes an experiment. In all experiments, a 4 m/s speed threshold and a 3 min maximum association time were used. The out-degree of each vertex was limited to 10.

We began our evaluation by calculating the H-measure for each experiment. For the purposes of the calculation, the degree-one edges were taken to be positive instances and the degree zero edges were taken to be negative instances. The misclassification cost ratio was set to the reciprocal of relative class frequency to control for the fact that there are significantly more negative class instances than positive. Table 5.1 shows the resulting values.

The first thing to note is the discrepancy between the performance of the two

	H-measure								
	5 fa/scan			15 fa/scan			30 fa/scan		
	$T=15\,\mathrm{s}$	$T=30\,\mathrm{s}$	$T=60\,\mathrm{s}$	$T=15\,\mathrm{s}$	$T=30\,\mathrm{s}$	$T=60\,\mathrm{s}$	$T=15\,\mathrm{s}$	$T=30\,\mathrm{s}$	$T=60\,\mathrm{s}$
Generative Tracklet									
CWNA/C1	0.981	0.966	0.843	0.983	0.920	0.880	0.960	0.924	0.820
CWNA/C2	0.981	0.966	0.843	0.983	0.920	0.880	0.960	0.924	0.820
CWNA/C3	0.952	0.883	0.767	0.928	0.851	0.761	0.910	0.841	0.621
PSH/C1	0.961	0.955	0.892	0.955	0.921	0.873	0.932	0.913	0.818
PSH/C2	0.961	0.955	0.892	0.955	0.921	0.873	0.931	0.913	0.818
PSH/C3	0.948	0.965	0.866	0.945	0.914	0.856	0.927	0.905	0.852
PSH/C4	0.947	0.965	0.866	0.945	0.916	0.856	0.926	0.906	0.853
PSH-ExpCorr/C1	0.981	0.971	0.871	0.976	0.915	0.829	0.950	0.922	0.842
PSH-ExpCorr/C2	0.984	0.971	0.872	0.976	0.915	0.829	0.950	0.924	0.843
PSH-ExpCorr/C3	0.984	0.971	0.872	0.975	0.915	0.829	0.950	0.924	0.843
PSH-Path/C1	0.977	0.994	0.969	0.970	0.968	0.951	0.955	0.963	0.935
PSH-Path/C2	0.956	0.981	0.923	0.953	0.950	0.918	0.940	0.930	0.949
Discriminative Tracklet									
Speed	0.376	0.566	0.683	0.305	0.482	0.602	0.352	0.496	0.643

(a) Constant $P_D = 0.9$.

Table 5.1: Tracklet model performance summary statistics.

	H-measure								
	5 fa/scan			15 fa/scan			30 fa/scan		
	$T=15\,\mathrm{s}$	$T=30\,\mathrm{s}$	$T=60\,\mathrm{s}$	$T=15\,\mathrm{s}$	$T=30\,\mathrm{s}$	$T=60\,\mathrm{s}$	$T=15\,\mathrm{s}$	$T=30\,\mathrm{s}$	$T=60\,\mathrm{s}$
Generative Tracklet									
CWNA/C1	0.940	0.893	0.820	0.929	0.874	0.751	0.911	0.852	0.693
CWNA/C2	0.940	0.893	0.820	0.929	0.874	0.751	0.911	0.852	0.693
CWNA/C3	0.902	0.817	0.724	0.877	0.759	0.666	0.843	0.777	0.596
PSH/C1	0.940	0.893	0.858	0.920	0.870	0.810	0.895	0.868	0.776
PSH/C2	0.940	0.893	0.858	0.920	0.870	0.810	0.895	0.868	0.776
PSH/C3	0.933	0.900	0.852	0.912	0.882	0.766	0.892	0.865	0.742
PSH/C4	0.933	0.900	0.854	0.912	0.882	0.766	0.892	0.864	0.742
PSH-ExpCorr/C1	0.941	0.891	0.817	0.919	0.870	0.757	0.898	0.857	0.747
PSH-ExpCorr/C2	0.941	0.891	0.818	0.922	0.871	0.758	0.899	0.860	0.749
PSH-ExpCorr/C3	0.941	0.891	0.818	0.922	0.871	0.758	0.899	0.859	0.749
PSH-Path/C1	0.977	0.955	0.961	0.962	0.941	0.889	0.938	0.917	0.887
PSH-Path/C2	0.973	0.981	0.942	0.963	0.967	0.917	0.928	0.945	0.928
Discriminative Tracklet									
Speed	0.373	0.576	0.666	0.362	0.586	0.624	0.354	0.515	0.638

(b) Random non-uniform binary $P_D = \{0.07, 0.93\}$.

Continuation of Table 5.1: Tracklet model performance summary statistics.

classes of tracklet models. As was mentioned already in the previous section, the discriminative tracklet model performed worse in terms of assigning a higher probabilities to degree-one edges. The generative models had H-measures that were > 0.9 for most experiments with worst performance being 0.596 for the notorious CWNA/C3 model at the highest false alarm rate and longest sample period. The average performance of the discriminative model, on the other hand, was closer to that level with the best result being only 0.683. Interestingly, the performance of the discriminative model actually improved as the sample period increased, approximately doubling between $T = 15\,\mathrm{s}$ and $T = 60\,\mathrm{s}$ across all false alarm rates and both P_D models. It is not clear why this should be the case as the straight line trajectory assumed by the model becomes more implausible as the sample period increases. At any rate, the results are a clear indication that when speed and probability of detection are the features, a generative model is preferable. We expect that the real advantage of using a discriminative tracklet model will come when incorporating high level information that would be infeasible to express in a tractable generative model. Given their performance, a more sophisticated discriminative model might wish to incorporate the output of one of these generative models as a feature expressing the consistency of data with a low-level kinematic description of the target.

Within the class of generative tracklet models, there were some surprises. Notably that the H-measure was not particularly well correlated with the accuracy of the underlying dynamic model. It is true that the least accurate dynamic model, CWNA/C3, also produced the lowest H-measure. However, the CWNA/C1 model outperformed the PSH/C4 model, with respect to the H-measure, in just over half of the data sets which is at odds with both the accuracy and consistency results. Also at odds with the accuracy and consistency results was the relative performance between the different configurations of the PSH- and PSH-Path-based models. Based on these results, we should use different dynamic models for the associative tracking model (feasible graph and inference) and any filtering pass on the sampler output intended to produce state estimates.

The observed discrepancies between the accuracy and the H-measure values in fact coincide with the differences in the underlying speed or heading model. It appears that when discrimination is the goal, the dynamic models based on the Wiener process have a slight advantage. Since each tracklet is the result of processing just two

measurements, the state initialization and process noise models are the primary factors that distinguish the models in this context. Unfortunately, it is not clear which is the dominant factor here. The position state is initialized in the same manner for all of the dynamic models. In the case of the Wiener process velocity models, the initial velocity state was estimated based on the second measurement using a two-point initialization scheme. For the other velocity models, that element of the state was initialized using the average speed for the given terrain.

Note that we do not see any differences in the results between the CWNA/C1 and CWNA/C2 models because they are in fact identical when applied to just two measurements; the same two-point initialization is used for both. We also fail to see any significant differences between the various PSH-ExpCorr-based models. This is likely due to the short correlation time relative to the interval between consecutive measurements because, when decorrelated, all three models are essentially equivalent.

For the data sets based on the non-uniform P_D model, the advantage of the Wiener process speed in the path-constrained model faded considerably. It is not clear why that is the case. In any event, for a given type of dynamic model, any improvement in the H-measure between the configurations with and without the Wiener process models is small; at most a couple of percent.

For the data sets based on the constant P_D model, the CWNA/C1 dynamic model produced one of the best tracklet models for short sample periods, even outperforming the path-constrained models in those cases. As the sample period increased, the advantage was lost and the path-constrained models produced the best results as expected. The CWNA/C1 enjoyed no advantage over the path-constrained models for the data sets based on the non-uniform P_D model, which we attribute to the additional missed detections which necessitate modeling the target's motion over a larger time interval much like with a long sample period.

Overall, the PSH-Path/C2 model provided the best discrimination with the difference being most pronounced in the most challenging environments. Even with $T = 60\,\text{s}$, 30 false alarms per scan, and the non-uniform P_D model, the H-measure was 0.928. The CWNA/C1 model, however, performed surprisingly well overall. This is welcome news because its computational complexity is the lowest among the models that we looked at.

To get some sense of what the different values of the H-measures mean in terms

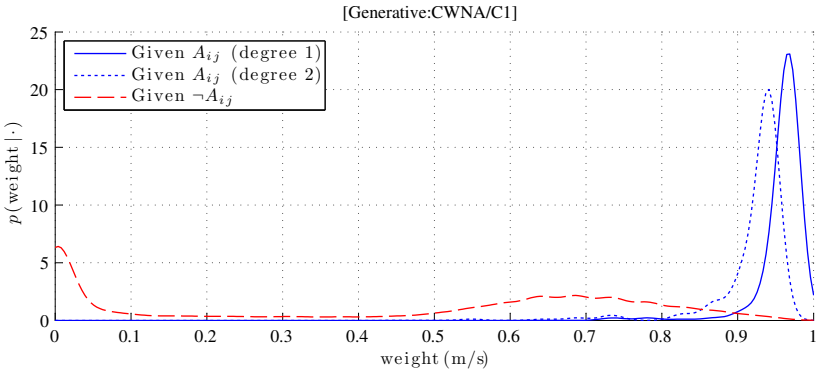
of the distribution of the edge weights, we estimated those distributions as well. Figure 5.5 shows a subset of those results taken from the experiments with $T = 15\,\mathrm{s}$, 15 false alarms per scan, and the non-uniform P_D model. The plots include the distributions of the degree zero, degree-one, and degree-two edges. Notice how that they are not quite what you might imagine if all you knew were the corresponding H-measure values. Despite the differences in their summary statistics, the weight distributions for PSH/C4 and PSH-ExpCorr/C2 look quite similar to that of PSH-Path/C2. The distribution of weights from degree-two edges for the PSH-Path/C2 model reveals another advantage compared to the other generative models, namely that expected weight values for the degree-one and degree-two edges are quite similar. However, the discriminative model can be seen to enjoy even more agreement between the distribution of weights from degree-one and degree-two edges. Of course, the plot also illustrates why the discriminative model suffers. There is a mode that is coincident with the mode from the distribution of weights from the degree-one edges.

Let us now look at how the ability of the edge weights to discriminate between true and false associations compares to the likelihoods that they are intended to approximate.

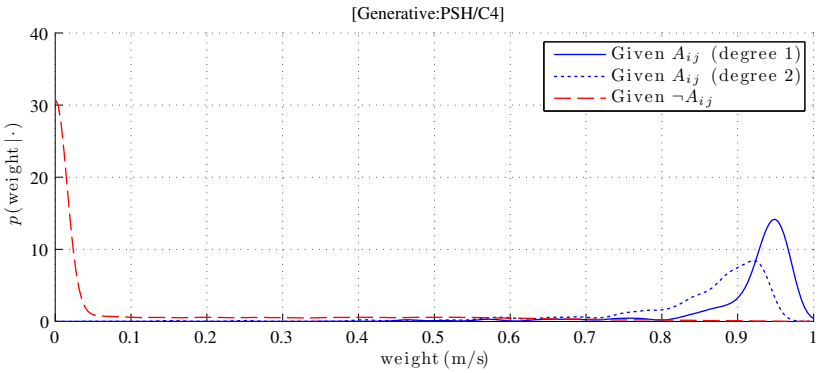
We performed this evaluation by calculating, for a range of lengths, the likelihood ratio between a true track and a sequence of invalid associations. The likelihoods were approximated using the tracklet probabilities by simply taking their product. For the generative tracklets, we repeated the likelihood ratio calculations using the standard likelihood model itself, with the corresponding dynamic model, in order to compare those results to the results obtained from the tracklet-based approximation.

A true track consists of a sequence of degree-one edges. Sequences of invalid associations were defined to be sequences of degree zero edges. Each ratio was calculated as the expected value of a random sample of such sequences from the feasible graph.

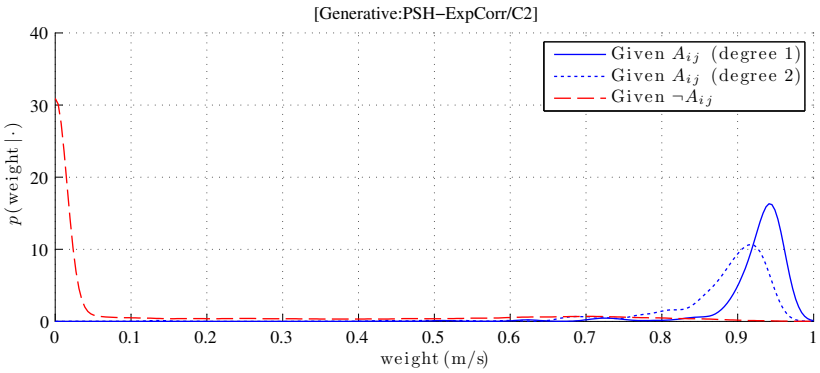
The results from these experiments confirmed that the likelihood approximations based on the tracklet probabilities are generally comparable to the true likelihood values in terms of discrimination. In some cases, the approximations are in fact superior. It should come as no surprise that the likelihood approximations based on the generative tracklet probabilities are comparable given the fact that those probabilities are computed directly from the standard likelihood model. Thus, the approximated likelihood in this case is just the true likelihood computed under the assumption of



(a)

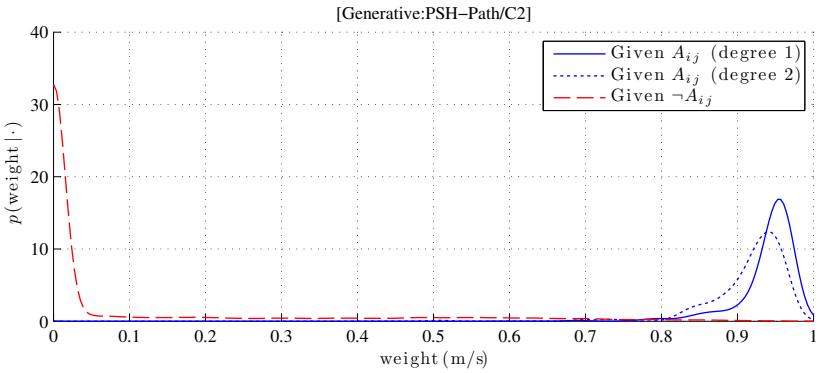


(b)

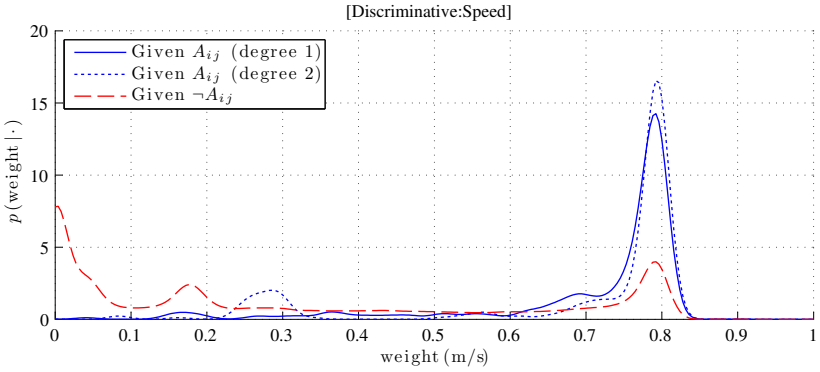


(c)

Figure 5.5: Distributions of edge weights for selected tracklet models.



(d)



(e)

Continuation of Figure 5.5: Distributions of edge weights for selected tracklet models.

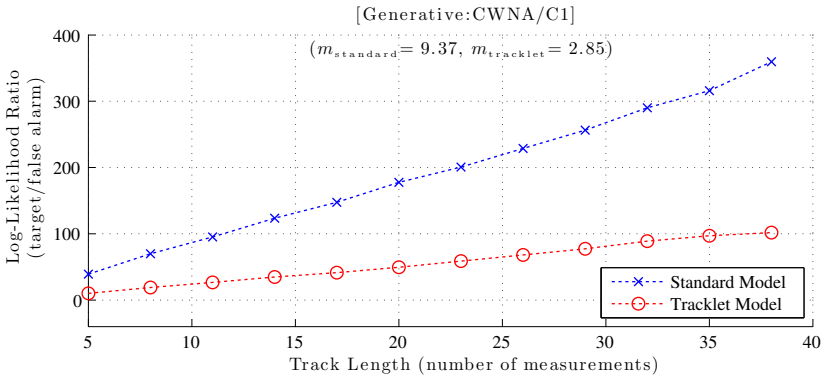
independence between measurements. The quality of such an approximation depends on how much information is lost when the dependencies are disregarded.

Since a concise summary of all the results is not possible, we have selected some representative examples to illustrate the important trends. We again focus on from the experiments with $T = 15\,\mathrm{s}$, 15 false alarms per scan, and the non-uniform P_D model.

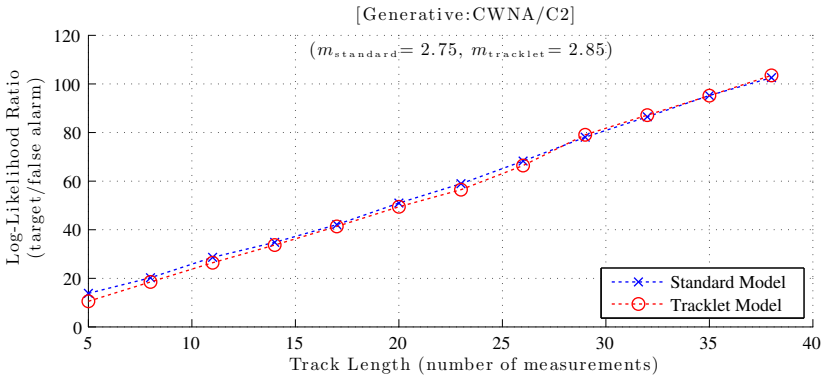
Within the class of generative tracklet models, any advantage of using dynamic models based on the Wiener process was largely lost. Across the board, such models were superior to the corresponding tracklet approximation in both absolute (i.e. the value of the log-likelihood ratio) terms and in terms of their growth rate with respect to the track length. However, they were generally inferior in these terms compared to the alternative configurations of the same dynamic model. The models that were not based on the Wiener process tended to exhibit much better agreement between the true likelihood and the approximation in both absolute terms and in terms of the growth rate. In some cases this represented more discriminative power than the alternative configurations based on the Wiener process models and in other cases less. The fact that the two were more closely matched is interesting because it may be beneficial to the sampler.

Figure 5.6 shows a comparison of the results for tracklets based on the CWNA model. The first configuration, which uses the Wiener process velocity model, is an example of particularly large deviation between the performance of the standard likelihood model and the approximation. With the second configuration, which employs the system rotation that effectively removes the preference for smoothness in heading, we can see the agreement between the log-likelihood ratios of the standard model and the tracklet-based approximation; the two curves are almost identical. It appears that the reorientation of the prediction at each time step effectively breaks the dependencies between the measurements in the joint likelihood. Comparing the results from the two configurations, it is clear that first has significantly more discriminative power. This, however, is not a typical result.

A comparison of the results for tracklets based on the PSH model is shown in 5.7. The first configuration uses Wiener process speed and heading models, and the performance of the standard likelihood model is again dominant here. Dropping the Wiener process heading model in the second configuration yields excellent agreement



(a)



(b)

Figure 5.6: Discriminative power of the likelihood approximation using tracklets based on the CWNA dynamic model.

as we saw above with the CWNA model. This too resulted in a loss of discriminative power. However, the fourth configuration dominates the performance of the other configurations. It is significantly better in both absolute terms and in terms of the growth rate. The agreement between the true likelihood and the approximations is still good here, but in this case the discrimination from the approximation is actually a bit better. This is a satisfying result for two reasons. First, PSH/C4 was one of the best performing dynamic models in terms of accuracy and consistency. Second, its computational complexity is relatively low which makes it a practical choice.

The results for the tracklets based on the PSH-Path model show similar trends to what we saw with the PSH-based tracklets. See figure 5.8. Here, the second configuration, which uses the stationary speed model, dominates the performance of the Wiener process speed model used in the first configuration.

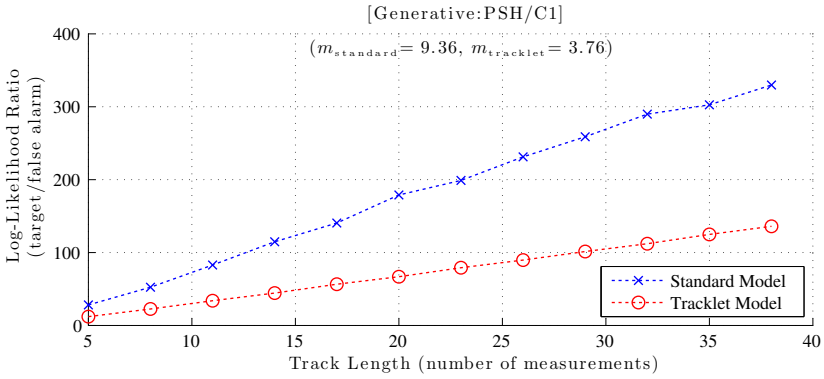
Finally, we consider the results for the tracklets based on the discriminative speed model. Figure 5.9 shows how the approximation performed. Here, there is no corresponding standard likelihood model with which to compare so we look instead at how slope and intercept compare to the other graphs. We can see that the discriminative tracklets had more discriminative power than any of the generative models that we looked at except for the PSH/C4 model. Indeed, the discriminative tracklet model is generally among the top performers with respect to the likelihood approximation. This is encouraging from the perspective of what we might expect from a future discriminative model that incorporates higher level information for which we do not have generative counterpart.

5.5 Inference

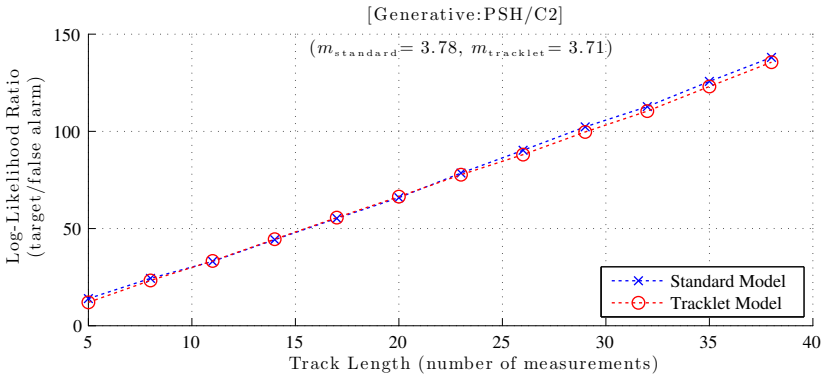
Ultimately, it is the inferences from the joint posterior:

$$p(Z,\, Z0,\, \lambda,\, \alpha \mid \tilde{Y},\, a_\lambda,\, b_\lambda),$$

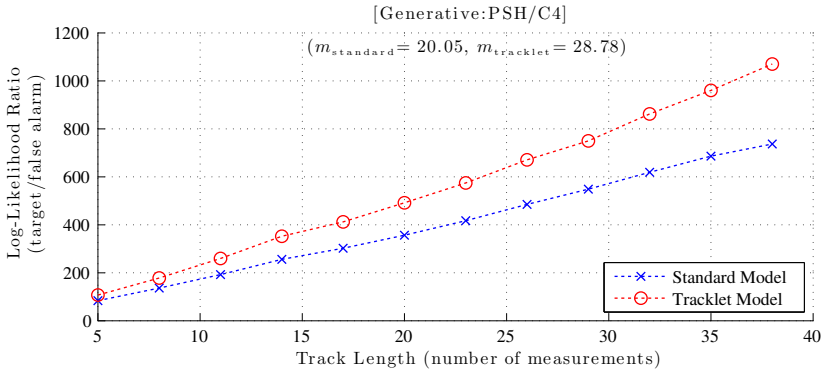
that we wish to obtain from the associative tracking algorithm. Using one of the algorithms described in chapter 2, MCMC or PMC, a sample of the posterior is drawn from which we can compute Monte Carlo estimates of the expectations of arbitrary functions of the indicator variables. These functions are chosen, based on the requirements of the specific application, to tell us all that we need to know about what is happening within the surveillance region.



(a)

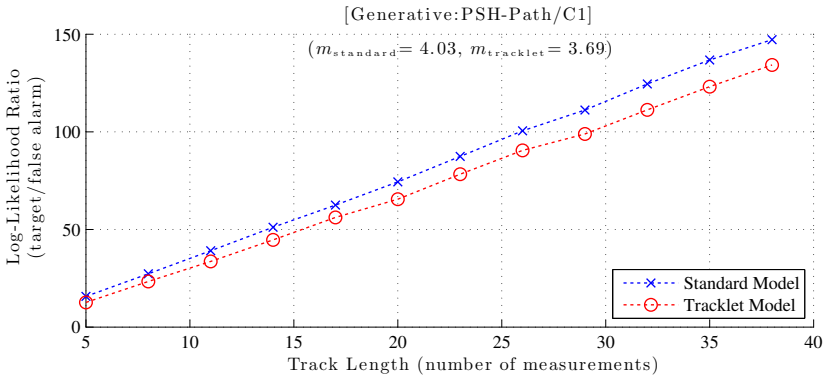


(b)

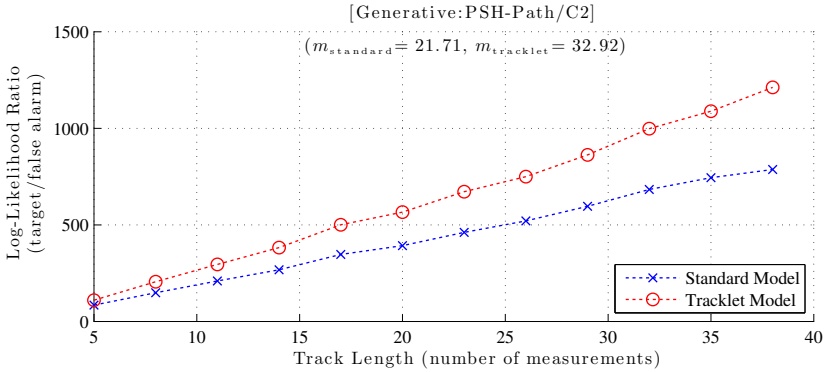


(c)

Figure 5.7: Discriminative power of the likelihood approximation using tracklets based on the PSH dynamic model.



(a)



(b)

Figure 5.8: Discriminative power of the likelihood approximation using tracklets based on the PSH-Path dynamic model.

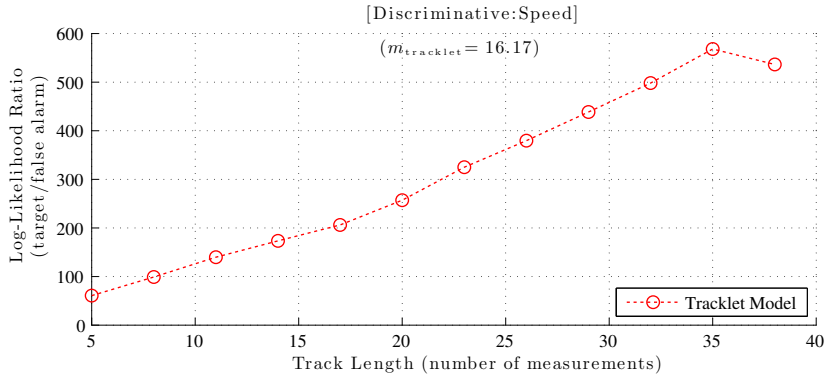


Figure 5.9: Discriminative power of the likelihood approximation using tracklets based on the discriminative speed model.

Whether or not a given sample will yield a suitable characterization of the scene primarily depends on two factors: how well the sample represents the underlying distribution and how well that distribution matches reality. The first is a function of the sampling algorithm while the second reflects the accuracy of the associative tracking model, which can itself be parameterized in various ways. Having already evaluated the dynamic models and feasible graphs, we assume that those choices are given. We further assume that the choices of false alarm model and probability of detection model are given as well. This leaves the prior distribution of the target cluster labels that we discussed in chapter 3 as the remaining free parameter in the associative tracking model. Thus, we can regard the second factor as a function of $p(Z \mid Z0, \alpha)$ and evaluate the effect of the different prior models here.

The quality of inference was therefore assessed for a given combination of sampling algorithm and target cluster label prior. For every such combination, the assessment was made by evaluating the resulting sample using a variety of means.

Unfortunately, the posterior sample cannot easily be visualized because of the large number of dimensions nor can the evaluation be distilled down to a single number. To complicate matters further, the extent to which the sample represents the underlying distribution cannot be measured directly. Regarding that first factor, we can only look for clues to how efficiently the sampler is exploring the state space. These clues include the MCMC acceptance rate, the sensitivity to initial conditions, and how the resulting MAP probability compares to the posterior probability of the true state (when that is in fact known). Given the enormous size of the state space and the fact that resource constraints will severely limit the sample size in practice, our primary concerns are that the sampler manages to find regions of high probability and does not get stuck near a mode that is unrepresentative of the true state. The focus of our evaluation was thus on the second factor because it is more amenable to quantification and the results shed some light on the performance with respect to the first factor as well. To evaluate the how well the distribution matched our knowledge of what was happening in the surveillance region, we used a collection of summary statistics.

The first summary statistic is simply the number of clusters, K. This conveys the most critical high-level information which is the answer to the question: how many targets are there? When processing simulated measurements, this value will be known exactly. Even in the case of measurements collected in the field, we should have a

good idea what to expect. The other summary statistics require ground truth, hence simulated measurements, because they are based on scoring the accuracy of the set of indicator variables.

For each cluster, we scored the accuracy of the corresponding indicator variables using the F-measure (or the balanced F-score)[15]. The F-measure is an evaluation metric that is commonly used in field of information retrieval to combine measures of precision and recall into a single statistic which takes values in the interval [0, 1], where indicates the best possible performance. For a given cluster label, precision is the fraction of the indicator variables that correctly identify the origin of the measurement. Recall is the fraction of the total number of measurements that originated from the corresponding target that were successfully identified. The two are combined by taking their harmonic mean. Despite its shortcomings for certain applications [149], we found the F-measure to be a convenient means of quantifying the notion that a "good" cluster should contain as many of the target measurements as possible while containing as few incorrect associations as possible.

To derive a combined score that accounts for all clusters, we essentially used the multiclass extension of the F-measure known as the macro-averaged F-measure [198]. This is computed in the same manner as the single class F-measure but using precision and recall values that are averaged across all classes. However, unlike the typical application of this technique, we have allowed the number of classes to vary with the input. This change was necessary to account for the fact that a given set of indicator variables may very well contain more cluster than the true number of clusters. We set the number of classes based on the maximum number of unique labels between the true and estimated indicator values. In other words, if the estimated Z contained fewer unique label values than the true number of clusters, then there was a class corresponding to each true target plus one for the set of false alarms. If, on the other hand, the estimated Z contained more unique label values than the true number of clusters, then there was a class corresponding to each true target, a class corresponding to each additional erroneous target, and one for the set of false alarms. This effectively penalizes sampler for generating excess clusters in addition to the obvious penalties for individual clusters that do not agree with the ground truth.

Until now, we have tacitly assumed that the correspondence between the cluster label values output by the sampler and those from the ground truth is known.

However, this is not true in general due to the exchangeability of Z with respect to the labels. To address this problem, we used the Munkres algorithm [128, 18] to determine the optimal mapping between the two sets of cluster labels.

While the scores convey useful information about how well a given (Z, $Z0$) describes the true association of measurements to targets, it can still be advantageous to attempt to visualize the distribution of the indicator variables implied by the sample. In cases where we lack ground truth, this is provides a basis for a qualitative assessment and is our only means of getting at the big picture. We chose to visualize the distribution of the indicator variables by using the sample to compute Monte Carlo estimates of the pairwise association probabilities. The result is essentially a colored version of the feasible graph where the color of each edge indicates its sample-based probability. Edges corresponding to associations with zero probability are not shown. Such weighted-edge-sample plots are used as appropriate to illustrate different samples.

Note that the weighted edge sample could also be scored to produce another summary statistic. However, it is not clear what an effective metric would be. The H-measure (and related metrics) are inadequate for this purpose as they will still yield large values when the non-track edge weights are unacceptably large but lower than the true-track edge weights.

For the samples generated using the PMC algorithm, the weights assigned to each element of provide another means of assessing the quality of the sample. We will monitor the variance of the weights, where a low variance is desired. We will also look at the effective sample size (ESS), which is given by:

$$\text{ESS} = \frac{1}{\sum_{i=1}^{n} w_i^2},$$

where w_i is the normalized weight of the i-th sample element and $1 \leq i \leq n$. The optimal value is $\text{ESS} = n$, although we must be aware that this can be achieved under less than ideal circumstances.

The false alarm rate, λ, and the concentration parameter (when applicable), α, are scalars so estimates of those variables can be reported directly. When ground truth is available, the expected false alarm rate will be known and the estimates can easily be verified. The concentration parameter, however, does not correspond to any measurable value and thus there is no truth with which to compare. We can view the estimates of α in light of its impact on the expected number of clusters to assess whether

or not they are reasonable. We can also compare the estimates to the most probable α according to its conditional sampling distribution (equation (2.33)) conditioned on the true set of indicator variables and the most probable value of λ (according to its conditional sampling distribution; equation (2.32)). That value will serve as a proxy for the true concentration parameter.

Each combination of sampling algorithm and cluster label prior under consideration was evaluated for a given data set. This constitutes an experiment. Three data sets were selected; two based on simulated measurements in order perform the evaluation with the benefit of ground truth and one consisting of actual radar measurements. The details regarding each of the data sets will be provided in the subsequent sections.

For the experiments based on the simulated measurements, the results of the sample evaluation are spread across two tables. The first table contains: the log posterior probability (denoted by LL), the combined score, the number of clusters, the concentration parameter, and the false alarm rate. This covers the key summary statistics as well as the estimates of α and λ. For each of these quantities, we reported the value corresponding to the sample element with the highest probability (i.e. the MAP value) and the value corresponding to the element with the maximum combined score. For all except the posterior probability, we reported the Monte Carlo estimate in terms of the expected value and its standard deviation as well. In addition, the true value was also reported for both the posterior probability and the concentration parameter. The true posterior was computed using the true (Z, $Z0$) and the most probable values of α and λ based on their respective conditional sampling distributions. The second table contains the scores for the individual true targets as well as the combined score, once again, for the sake of completeness. For each of these fields, we reported the scores corresponding to the MAP element, the element with the maximum combined score, and the Monte Carlo estimates (in terms of the expected value and its standard deviation).

When ground truth was not available, a single table was used that contains: the log posterior probability (denoted by LL), the number of clusters, the concentration parameter, and the false alarm rate.

The tables for all experiments are collected in appendix D.

5.5.1 Sampling Algorithms

In section 2.4, we discussed two different algorithms for generating a sample from the posterior distribution: MCMC and PMC. Both are based on the same mixture proposal distribution that was detailed in section 2.4.3. In addition, we suggested an alternative mixture proposal distribution (see section 2.4.3.3), composed of a set of k-add moves, intended for use with the PMC algorithm. All three approaches were evaluated here.

5.5.1.1 Move Definitions

Recall that the mixture proposal distributions are based on a weighted combination of reversible moves. Each move can be defined in a number of different ways and in chapter 2 we described our preferred approach along with some alternatives. Let us now mention some of what was tried in preparation for these experiments and describe the final configurations that were used.

Being the most complex and the most critical to the performance of the sampler, we tested several variations of the add move. We began with the approach of selecting one of the false alarm vertices to initiate the path, selecting the length, and then constructing the path. It had the appeal of being the least computationally demanding implementation that we considered. Unfortunately, in our preliminary investigations it performed poorly compared to the more complex approaches in which the starting measurement itself is selected with consideration for the possible paths. The resulting sample tended to overestimate the expected number of targets and yield a low expected combined score. In some runs, the acceptance rate suffered because of rejected add proposals. However, the basic principle of first selecting the length of path, followed by the systematic selection of vertices from among paths of the chosen length, proved to be a very effective alternative to the traditional approach of randomly selecting each vertex until some stopping criteria is met or, as is often the case, there are no more feasible additions. It avoids the tendency to produce short paths that are not plausible tracks. The more complex approaches to implementing the add move take this a step further by selecting the length of the path prior to selecting the initial vertex. The initial vertex is then selected based on average weight of a path of the chosen length. These approaches tended to yield long, plausible paths and thus became our focus.

In practice, calculating the weights by running the estimation procedure repeatedly for each false alarm vertex turned out to be too slow in our unoptimized implementation. Although that method is amenable to parallel computation, we settled instead on one based on running the estimation procedure for each source vertex. The latter relies on an approximation to the ideal weight calculation that proved to be acceptable in practice. We did consider drawing the path length uniformly at random or with a bias toward longer paths, rather than using a probability proportional to the aggregated weight for that length, but this led to lower MCMC acceptance rates and paths that were more likely to consist of noise. Finally, in order to encourage more diverse, but still promising, add proposals, we used a simple modification of the path construction procedure for these experiments. We drew the length and selected the first vertex using probabilities based on the aggregated weights. The remaining vertices, however, were selected uniformly at random. Note that because of the way the bookkeeping is done when the weights are calculated, there will always exist at least one path from the chosen starting vertex of the desired length.

When the cluster to be deleted is selected with a probability that is proportional to the weight of the selection, the sampler tends to favor states that compare well with the ground truth. However, this is perhaps too much of a good thing. This selection procedure causes the sampler to linger around certain modes of the distribution longer than we might like. The simple alternative of selecting the cluster to be deleted uniformly at random proved to be more effective. It also brought a welcome reduction in the computational requirements of the algorithm. When the cluster deleted happens to be part of a configuration with significant weight, with a high probability it, or something close to it, will be reconstituted by the next add move. Thus, the sampler can return to such high probability states very quickly because of the type of add moves that we employ.

A delete move, with the cluster selected uniformly at random, followed by an add move is exactly what we defined as a replace move. The idea was to provide an additional means of modifying the state in a way that does not alter the number of clusters. Unlike the extend, reduce, or switch moves, the replace move has the potential to propose a candidate state in which the replacement cluster has no vertices in common with its predecessor. Thus, it facilitates large but meaningful transitions in a single step. We found it to be extremely effective in practice, leading to superior mean scores and

MAP estimates. It does, however, increase the complexity of the PMC sampler. At issue is the (first) Rao-Blackwellization step. When the moves do not overlap, meaning that given the current state two different moves cannot propose the same candidate state, the normalization constant is trivial to calculate. Unfortunately, the replace move will, with some nonzero probability, propose the same state as any given extension or reduction. When computing the normalization constant, this should be accounted for. However, we found little difference in practice even when the additional contribution to the normalization constant was simply ignored.

You might have noticed at this point that we are relying on the add move to do much of the heavy lifting. We could take that a step further and eliminate the extend/reduce moves, relying on the replace move to achieve that effect. Motivated in large part by the fact that it removes the aforementioned complication with the Rao-Blackwellization step without having to forgo the benefits of the replace move, this approach was tested. The change reduced the variance in the expected number of clusters but also reduced the tendency of the sampler to explore clusters with lower likelihoods. We also considered the impact of eliminating the split/merge moves; again to rely on combinations of adds and deletions to achieve the same effect. The results were similar to what we saw after eliminating the extend/reduce moves. However, proposal distributions lacking either the extend/reduce or split/merge were deemed to be too close in spirit to the k-add proposal distribution, which relies entirely on the add move (with the necessary deletions being implicit), to warrant further study.

Regarding the extend and reduce moves, we tried one variation on the original approach. Instead of selecting the cluster to extend, or reduce, using a probability based on the weights, it was selected uniformly at random. This seemed to just exacerbate the problem of target clusters that contain noise. The experiments were thus conducted using the original weighted selection.

The split move was implemented using the approach in which the cluster to split was selected uniformly at random. This was primarily based on computational considerations. The more expensive approach based on enumerating all possible split points was not implemented.

No alternatives to the original merge or switch moves were considered.

Impacting all of the moves are the contributions to the weights. Recall that those come from the four factors in equation (2.38). Ideally, we want to include them

all, but we first sought to confirm their effectiveness. As expected, the contributions from the measurement likelihood, prior distribution over the target cluster labels, and the prior distribution over the false alarm cluster labels were all critical to achieving good performance. The contribution from the concentration parameter hyperprior, however, was not found to be effective. For cluster label priors that respect the scan constraints, that contribution cannot be computed exactly and the approximation is still quite expensive. Since it appears to provide no discernible benefit, at considerable cost in many cases, the contribution from the concentration parameter hyperprior was not used. Regarding the measurement likelihood, we tried three different ways of computing its contribution. The first was based on the product of the complementary edge probabilities. In other words, edges between vertices assigned to the same target contribute $\log(p)$ and all other edges contribute $\log(1-p)$. The second approach used a small constant probability for all non-track edges so that the model would more closely approximate the uniform false alarm model. Finally, the third approach was based on a modified version of the feasible graph in which the edge weights were the joint measurement likelihood of the pair. These were combined with the false alarm likelihood to generate an approximation to the full measurement likelihood based on the assumption of pairwise independence within each cluster. This, of course, entails the abandonment of the notion of the discriminative tracklet probability and any of its potential benefits. It turned out that the first approach was the clear winner with the second approach leading to very poor results. The third caused the sampler to explore much less of the space as evidenced by a lower MAP probability and tendency to completely miss tracks with lower likelihoods. These experiments were thus conducted using the product of the complementary edge probabilities to approximate the likelihood contribution.

5.5.1.2 Proposal Distribution Definition

Having described how the individual moves were implemented, something more needs to be said about how they are combined to form the mixture proposal distribution because here too we looked at an alternative to what was presented in chapter 2. The following applies to the mixture proposal distribution that is shared by the MCMC and PMC samplers.

Rather than constructing the proposal distribution as a mixture of all the different moves as described in section 2.4.3, we considered a mixture three meta-moves.

Each meta-move comprises the individual moves that alter the state in a similar manner. The first contains the add and split moves, which alter the current state by increasing the number of clusters by one. The second contains the delete and merge moves, which alter the current state by reducing the number of clusters by one. Finally, the third contains the switch, extend, reduce, and replace moves, which alter the current state in a way that retains the number of clusters. Within each meta-move, the move used to actually generate the proposal is selected uniformly at random. Now, from the perspective of the MCMC sampler, this is equivalent to particular choice of mixture weights is not particularly interesting. However, it offers a small practical advantage. If we wish minimize the chance that a proposal fails because the selected move is not possible given the current state, then we have to check the feasibility of each move and rescale the weights if necessary. Using the meta-move mixture proposal distribution reduces the amount of checking required for each draw. However, the savings is small and the real motivation behind the change was the weight adaptation of the PMC algorithm. During the course of our testing with the default mixture proposal distribution, we observed that the weight adaptation frequently failed offer any improvement over runs using fixed uniform weights. This was despite the fact that we used Rao-Blackwellization to decondition the chosen kernel, which was originally introduced into the D-kernel PMC algorithm to address the problem of the weight adaptation failing to learn from experience. Although we do not fully understand why, the meta-move mixture proposal distribution performs much better in this regard. Therefore, it was utilized for these experiments. Future work will have to determine the failure mode.

Regarding the weight adaptation in the PMC algorithm, recall that the idea behind the k-add mixture proposal was to use the adaptation scheme learn the distribution of the number of targets. Here, it performed as expected using the proposal distribution in its original form.

5.5.1.3 Status Quo

Something else that impacts all of the moves, but has remained unmentioned until now, is whether or not they include the current state as a potential candidate state. This is an important question whose answer significantly affects the performance of the sampler. It was not fully appreciated when the model was originally conceived and we have delayed the discussion in order to undertake it from the perspective of modifying

the sampler as presented.

With the exception of the replace move, none will produce a candidate state that is identical to the current state. Implicit in their respective definitions is that the candidate state is selected from among the feasible moves without explicitly considering the option of doing nothing. For example, if there is at least one target cluster and the delete move is valid, then it will always propose to move to a state with one fewer target cluster. Existing MCMC data association algorithms work in exactly this manner (e.g. [135]). However, it is sometimes desirable to include the option of maintaining the status quo. Or, in our case, provide mechanism for proposing to keep the current state other than a replace move that happens to regenerate the original cluster.

There are two general approaches we might take. The first would be to have each move consider the current state in addition to the other feasible candidate states. For example, an add move would consider paths of length zero and a delete move would consider not removing any clusters. The second approach would be to keep the moves as defined but add a kernel to the mixture that represents a null move which has no effect on the current state. This is something that we see in non-tracking applications of MCMC and PMC sampling, even when the support of the individual kernels covers the entire sample space. In our implementation, we would add the null move as a null meta-move in the mixture proposal distribution. For the PMC algorithm, the weight adaptation scheme would then determine how frequently to use it.

We implemented both approaches and were unable to settle the debate in our initial investigation. The default implementation, which did not explicitly entertain keeping the status quo, clearly explores more of the space as evidenced by the higher variance in the expected number of clusters and the higher MAP probabilities. However, it generally yielded lower expected scores (both per-track and combined); especially when it comes to the score corresponding to the MAP estimate. This suggests that there may be a practical issue of quality versus quantity in how the sampler moves through the state space. Given the fact that most applications will be limited to a relatively small sample size (compared to the size of the state space), it is critical that the sampler move efficiently. Ideally we want it to visit configurations that agree with the true state most of the time without simply getting stuck there. It appears that having each move consider selecting the current state leads to a good tradeoff in this regard. For that reason, all experiments (except those based the k-add mixture proposal) were

run using both approaches to dealing with the status quo. No changes were made to the k-add moves since they already allow for the possibility of proposing the current state. The option of increasing the odds of such a move through the introduction of a null move to the k-add mixture proposal was not considered.

5.5.2 Cluster Label Priors

In chapter 3, we discussed a number of different ways to define the prior distribution of the target cluster labels, $p(Z \mid \mathcal{Z}0, \alpha)$. In these experiments, we evaluated them all including several variants of each of the nonparametric priors.

Recall that the nonparametric priors were the DP, UP, EP, and EP2. They are all functions of a concentration parameter, α, whose value can be specified in various ways. We adopted the Bayesian approach of treating it as a random variable, with its own hyperprior, to be included in the inference process. The hyperprior, $p(\alpha \mid \mathcal{Z}0)$, is defined by two parameters of its own. Those were set to $\eta = 1 \times 10^{-5}$ and $minClusterSz = 7$. As we described in section 3.5.1, the specific form that it takes depends on whether or not the moments, $E[k \mid \alpha, n]$ and $Var[k \mid \alpha, n]$, are estimated subject to the scan constraint. We included both the constrained and unconstrained forms in our experiments. In addition, we also included variants in which the concentration parameter was assumed to be known and constant. A value of $\alpha = 1 \times 10^{-6}$ was used to favor the smallest number of clusters possible. Note that this effectively removed the concentration parameter hyperprior from the model.

The HUP and the scan-constrained HUP have no parameters and were implemented as described.

Three different property-based priors were considered. The first simply expressed a preference for a small numbers of clusters. It will be referred to as the "K" prior. The distribution was defined using exponential-type decay model for $p_0(K \mid \mathcal{Z}0)$ from section 2.3.2.2 with $\eta = 1 \times 10^{-6}$. As we illustrated in section 3.4.2, the strength of this preference is not sufficient to overcome the structural deficiencies inherent in this prior. However, since it represents a well-intentioned attempt, we let it stand to serve as an example of a prior that does not represent the intended beliefs. The second property-based prior expressed both a preference for long tracks and pairs of target measurements that are close in time. This will be referred to as the "Prop." prior. A growth variant of the $p_0(K \mid \mathcal{Z}0)$ model with $\eta = 1 \times 10^{-4}$ was used to quantify length

preference where the minimum length was taken to be 2 and the maximum length was equal to the number of scans. The temporal proximity preference was quantified using the exponential-type decay model, defined over scans, with $\eta = 1 \times 10^{-2}$, to weaken the preference in light the non-unity probability of detection. The minimum gap was one scan and the maximum was set to the association window size of 3 minutes. The third property-based prior was that of Oh et al., which was described in section 3.4.2. It will be referred to as the "Oh" prior. There are three parameters that must be specified: p_d, p_z, and λ_b. Recall that these are the probability of detecting a target during a given scan, the probability of a target disappearing during a given scan, and the birth rate of new targets per scan per unit volume (or area), respectively. Only p_d has a known value, which is equal to the expected probability of detection over the surveillance region. Since the other parameters require knowledge about the number and lifetimes of the targets that we do not have, we treated them as independent random variables with their own hyperpriors.

For the probability of disappearance, we assumed that an approximately uniform prior was appropriate, but with a slight preference for lower values. To represent these beliefs, we modeled $p(p_z)$ using a beta distribution with $\alpha = 1.0$ and $\beta = 1.1$. For the birth rate, we assumed that small values are more probable than a large ones and that an exponential-type decay was an appropriate form. The motivation was to remain consistent with how we represented our beliefs about the number of targets in $p_0(K \mid \mathcal{Z}0)$. Here, we used a continuous variant of the decay model from section 3.5.1 in order to again set the rate based on a ratio of the largest to the smallest value. The resulting distribution is given by:

$$p(\lambda_b \mid \mathcal{Z}0) = -a^{\lambda_b} \log a,$$

where $a = \eta^{\frac{1}{\lambda_{b_{max}}}}$ and $\lambda_b \leq 0$. Just like the concentration parameter hyperprior, this one is defined by the η and *minClusterSz* parameters. The decay rate is determined by η, which specifies the ratio of the densities at the maximum birth rate, $\lambda_{b_{max}}$, and the minimum birth rate, zero. The maximum birth rate is determined by setting an upper bound on the number of clusters, which is given by:

$$K_{max} = \left\lceil \frac{nm_{\tau_+}}{minClusterSz} \right\rceil,$$

where nm_{τ_+} denotes the number of target measurements. The maximum birth rate is then given by: $\lambda_{b_{max}} = K_{max}/ns^+$, where ns^+ denotes the number of scans that contain

at least one target measurement. Note that both ns^+ and nm_{τ_+} are known from the conditioning on $\mathcal{Z}0$. The η and *minClusterSz* parameters were set to the same values that we used for the nonparametric priors.

5.5.3 General Remarks

All of the MCMC experiments were conducted using just 7000 samples. For the PMC experiments, in order to maintain an equivalent allocation of resources, a sample size of 1000 with 7 resampling iterations was used. Given the enormous size of the space, this is clearly insufficient for a thorough exploration. However, we did not consider using a larger sample because real tracking systems will not have that luxury without significant increases in computing power. One of the questions that this research sought to answer was what we can expect from a small, but hopefully well-chosen, sample.

Given the resource constraints, the asymptotic justification of MCMC sampling is arguably insufficient and there is no reliable method for establishing convergence. This motivated us to consider PMC sampling which relies on importance sampling arguments that are more convincing under these conditions. Another question that we sought to address was how these two approaches compare when used to generate a small sample.

The initialization scheme described in section 2.4.3.4 proved to be quite effective, generally producing high scoring initial states. However, its effectiveness did vary somewhat depending on the cluster label prior. This is because the impact of the prior on weights used by the add move is accounted for to varying degrees. Depending on the accuracy of the weight estimates and the strength of the prior, the initial number of clusters can differ significantly. So, for the sake of consistency, we opted instead to use a different approach. In the MCMC experiments, the initial state was simply set to all false alarms. In the PMC experiments, we drew the initial state for each element in the sample from a proposal distribution based on repeated add moves. Unlike the initialization scheme that we decided to pass on, this distribution selects the number of clusters uniformly at random, up to a predefined maximum value, and thus does not rely on drawing an add move of length zero to determine the total number. The idea was encourage a diverse sample, with elements consisting of different numbers of clusters. A maximum of 25 clusters was used for these experiments.

To set the parameters, a_λ and b_λ, for the false alarm rate hyperprior, we

adopted the convention of fitting a gamma distribution to the number of measurements in each scan of the data to be processed. This assumes that all of the measurements are false alarms, which may or may not be true. For the one target data set, the resulting estimate was quite close to the true value. On the other hand, this procedure results in an overestimate when applied to the five target data set. The idea was to use a consistent procedure that assumed no prior knowledge of the true rate. In practice, we expect that obtaining accurate estimates of the false alarm rate will not be a problem. Here, we deliberately accepted a poor estimate in the case of the five target data set to see how that affected the final outcome.

Regarding the PMC algorithm, resampling was performed using the stratified resampling algorithm. Contrary to the results of Hol et al. [84], we did not find the systematic resampling algorithm to be preferable. In our experience, it showed a slight tendency to produce more samples that were dominated by a single element (i.e. particle) than the stratified resampling algorithm. The second Rao-Blackwellization step was not implemented due to cost concerns.

Finally, the two data sets based on simulated measurements were generated using the same scan period and false alarm rate; $T = 30\,\mathrm{s}$ and 15 false alarms per scan, respectively. These values were selected as they represent the midpoint from the feasible graph evaluation. As before, we used a randomly generated binary model with $P_D = \{0.93, 0.07\}$; the low probability of detection regions cover approximately one quarter of the surveillance region.

5.5.4 One Target

This data set consists of a single target walking in rolling terrain approximately 5 km southwest of the sensor. For the first 17 minutes, the target has not yet appeared in the surveillance region so the tracker must contend with a steady stream of false alarm measurements. After the target does appear, it persists for approximately 15 minutes at which point the data set ends. There is a mix of in-range and cross-range motion. Figure 5.10 shows the P_D along with the detections and true track. Notice how the targets passes through several small regions of low probability of detection.

The feasible graph was constructed using the PSH/C4 model with the maximum out-degree of each vertex limited to 5. This yielded complete coverage of the degree-one edges. It turned out, however, that PSH/C4 model was not the optimal

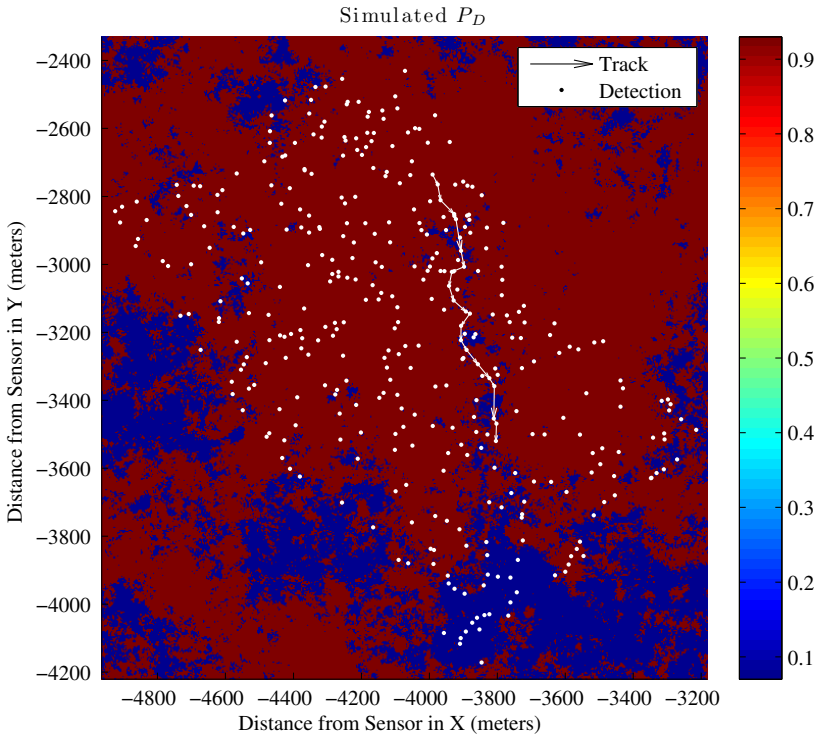


Figure 5.10: Simulated P_D with the detections and tracks for the single target data set.

choice for the dynamic model used to specify the measurement likelihood. As we saw earlier while evaluating the edge weights of the feasible graph, dynamic models based on the Wiener process have an advantage when it comes to discrimination because of their implicit preference for smoothness. This advantage became quite apparent during our study of the inference process. Using the PSH/C1 model for the measurement likelihood produced substantially better results and was thus adopted for these experiments. Compared to the PSH/C4, the resulting samples contained far fewer high probability tracks consisting solely of false alarm measurements. The inferior state estimation of the PSH/C1 was noticeable when logging out of range queries to the P_D model, which were mapped to a negligible probability of detection. By comparison, no such out of range queries were encountered when using the PSH/C4 model.

Tables D.1–D.10 contain the summary statistics from the experiments.

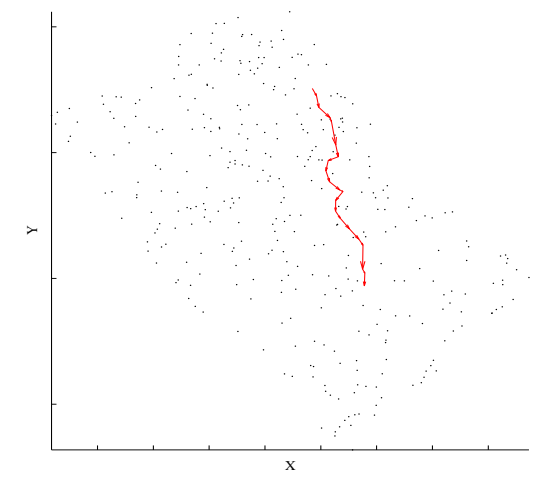
Looking first at the performance of the MCMC sampler, we begin by noting that the experiments using the default implementation of the proposal distribution revealed a tendency to grossly overestimate the number of targets present. The estimates of K also contain considerable variance which is a discouraging prospect for surveillance applications. The estimates range from a very reasonable 1.8 ± 0.40 with the DP prior using a small fixed α to the preposterous 53.9 ± 12.08 produced by the "K" prior. That last result is perhaps unsurprising as the "K" prior was meant to serve as an example of a poorly chosen prior. However, in these experiments, the EP prior performed almost as poorly, with several others not far behind. Only those priors that expressed a strong preference for a small number of clusters, the nonparametric priors with a small fixed α and the "Prop." prior, managed to keep the estimated number of targets to single digits. The false alarm rate is underestimated to varying degrees as we would expected given overestimates of the number of targets.

For the nonparametric priors, the high numbers of clusters correspond to high values for the concentration parameter. Note that the truth values for α in table D.1 are all the same because 1×10^{-6} was used as the lower limit of the support for the grid approximation of equation (2.33). This should be understood to mean that the true value is $0 < \alpha \leq 1 \times 10^{-6}$. In retrospect, our choice for minimum should probably be reconsidered in the future. Here, we can see how the scan-constrained versions of the nonparametric priors indeed produced lower expected α values. Although the impact of the scan constraints is imperfectly accounted for, the compensation is clearly

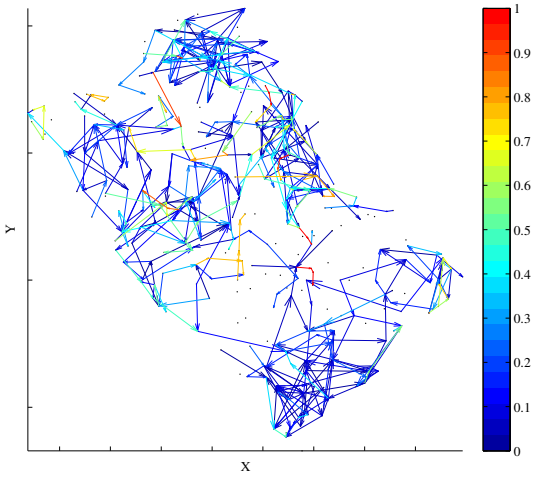
beneficial. As an example, consider the DP prior. The scan constraints incorporated into the hyperprior reduce the MAP estimate of the concentration parameter from 14 to 6.86, which corresponds to 49 and 22 targets, respectively. Using a small fixed value, $\alpha = 1 \times 10^{-6}$, drove the MAP estimate of the number of targets down to just two, with one of the two being almost a perfect match to the ground truth. Unfortunately, it is not clear exactly why the scan constraints did not drive the concentration parameter down further. It might be inaccuracies in our accounting for the impact of the constraints or the beliefs used to define the hyperprior might place too much weight on larger numbers of clusters. With the first potential cause less amenable to improvement due to its inherent complexity, the second is a more promising avenue for further study. Figure 5.11 shows the weighted-edge-sample plots for each case. Although it may a bit difficult to discern in the plots, the unconstrained prior has the most erroneous edges occurring with probability > 0.5. The scan constraints have improved the situation in that regard but at the cost of lower weights for the true-track edges. Lowering α to express a strong preference for the lowest possible number of clusters produced a very accurate picture of what was really going on. This suggests that future work should revisit the definition of the hyperprior with an eye toward strengthening the preference for small values of the concentration parameter.

One might wonder if we could reduce the tendency to overestimate the number of targets by enforcing a minimum cluster size. This would effectively further[2] truncate the posterior distribution by placing zero prior probability on all configurations containing clusters with insufficient size. We chose not to address this question here in order to both limit the number of variables in the experiments and establish a performance baseline without the constraint. However, an initial investigation suggests that the approach is promising. Figure 5.12 shows an example of the results obtained by setting the minimum cluster size to 8. It was implemented by modifying the moves in the proposal distribution to enforce the constraint when computing the selection probabilities. The DP prior was taken as the example so that the weighted-edge-sample plots can be compared to those in figure 5.11. The expected number of clusters, given the minimum cluster size, was 8.2 ± 1.69, 7.2 ± 2.09, and 1.8 ± 0.41, for the unconstrained, scan-constrained, and small fixed α variants, respectively. The respective expected combined scores were: 0.18 ± 0.089, 0.21 ± 0.10, and 0.71 ± 0.13. For the unconstrained and

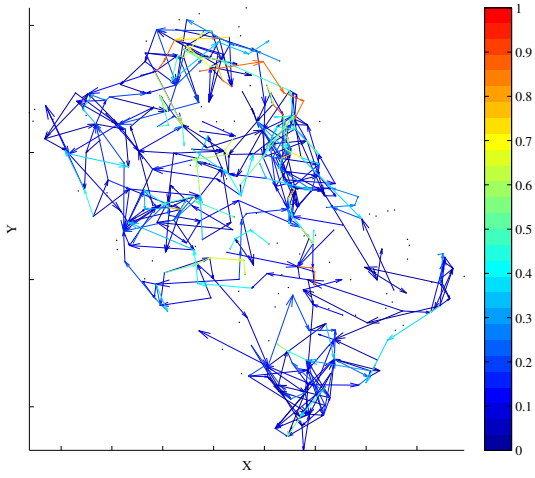
[2]Further than the implicit truncation from the scan constraints and maximum association time.



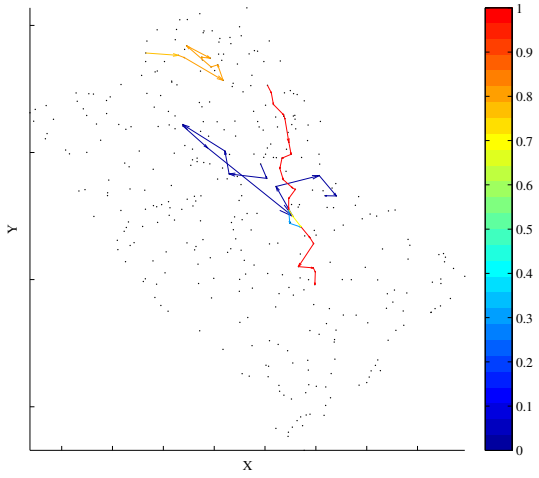
(a) Truth.



(b) Sample based on the unconstrained DP prior with a mean α of 12.4.



(c) Sample based on the constrained DP prior with a mean α of 4.81.



(d) Sample based on the DP with $\alpha = 1 \times 10^{-6}$.

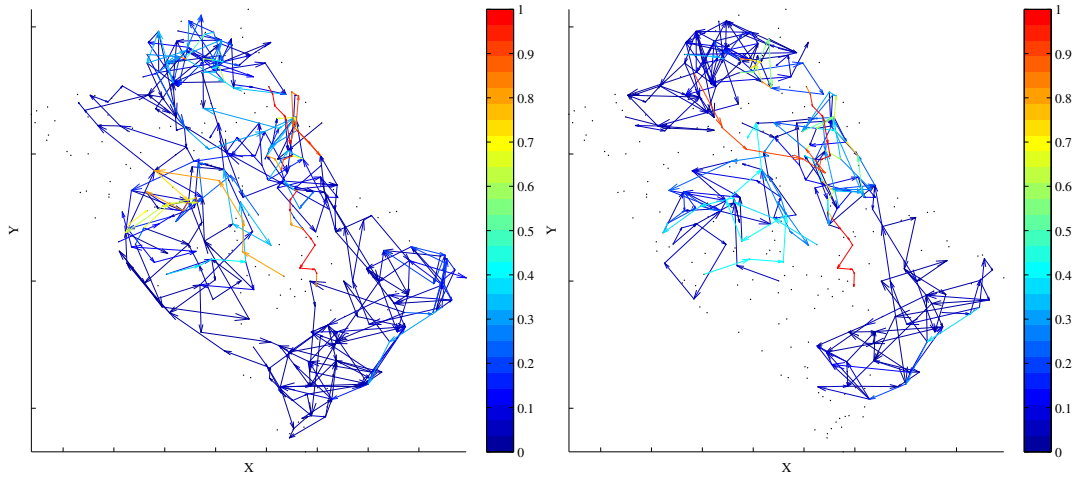
Figure 5.11: Example of the effect of scan constraints applied to the DP prior. The results were derived from an MCMC sample of the one target data set.

scan-constrained versions of the DP prior, the estimates have clearly improved.

The posterior probabilities reveal just how critical the choice of cluster label prior is to getting a posterior distribution that matches reality. Notice how accurate the maximum-score point estimates are in table D.1. This is true across the board; independent of the prior. These are exactly the kind of results that we were aiming for, yet in most of the experiments the sampler spent relatively little time there, or near similar configurations, because they were considered much less likely according to the model. Thus, the generally poor results here are not caused by the sampler failing to explore the highest probability regions of the space but are rather the result of those regions not being what we want them to be. The gap between the posterior probability of the true and MAP-estimated variables reveals a lot about how good the model is. When the gap is large, the model is assigning high probabilities to configurations of the indicator variables that are unlike the true state. At least with this data set, only a prior distribution of cluster labels that expresses a very strong preference for the smallest possible number of clusters will yield a suitably accurate model.

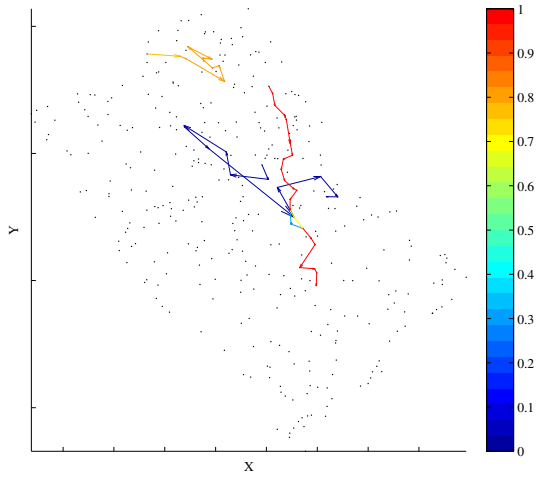
It was somewhat surprising to see that the DP prior performed best despite its tendency to favor uneven distributions of cluster sizes. We had expected this preference to undermine its performance by placing too much weight on configurations with numerous short tracks consisting entirely of noise. This led us to propose the EP and EP2 in addition to considering the more neutral UP. However, in practice, the "rich-get-richer" behavior of the DP offered a distinct advantage: it resisted the breakup of large clusters into more or less equally sized pieces. Figure 5.13 shows a comparison of the MAP estimates of Z from the nonparametric priors with a small fixed α. Notice how each one represents the kind of configuration that we might expect knowing the properties of the underlying stochastic process. The estimate corresponding to the DP contains two clusters of unequal size. The UP estimate contains a relatively uniform distribution of cluster sizes. Finally, the EP and EP2 estimates contain clusters that are all nearly the same size. Unfortunately, for the EP and EP2 priors this came at the expense breaking up the one true track.

The advantage of the DP prior in these experiments can also be seen in the distribution of cluster counts implied by the sample. In figure 5.14, we can see the distributions of the number of clusters resulting from the different scan-constrained nonparametric priors. Only the scan-constrained DP prior has a significant mode at the



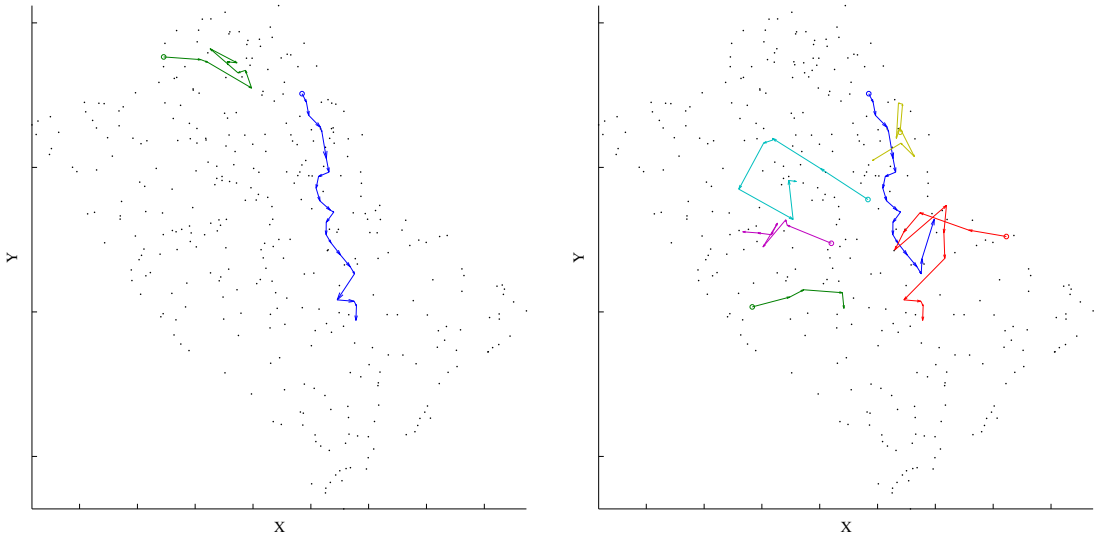
(a) Unconstrained DP prior.

(b) Scan-constrained DP prior.



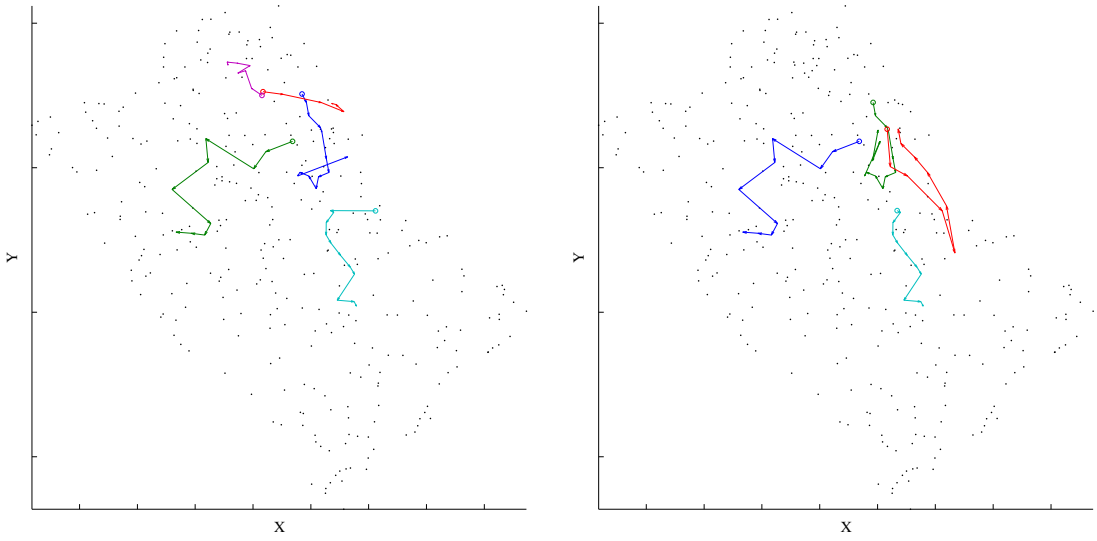
(c) DP prior with $\alpha = 1 \times 10^{-6}$

Figure 5.12: Example, using the DP prior, of the effect of setting the minimum cluster size set to 8. The results were derived from an MCMC sample of the one target data set.



(a) DP prior with $\alpha = 1 \times 10^{-6}$.

(b) UP prior with $\alpha = 1 \times 10^{-6}$.



(c) EP prior with $\alpha = 1 \times 10^{-6}$.

(d) EP2 prior with $\alpha = 1 \times 10^{-6}$.

Figure 5.13: Comparison of the MAP estimates of Z from the nonparametric priors. The results were derived from an MCMC sample of the one target data set.

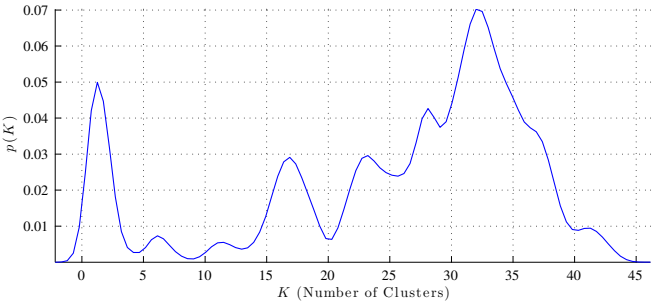
true value, $K = 1$.

See figure 5.15 for the weighted-edge-sample plots from the scan-constrained nonparametric priors that we have not already looked at; the plot from the scan-constrained DP is in 5.11c.

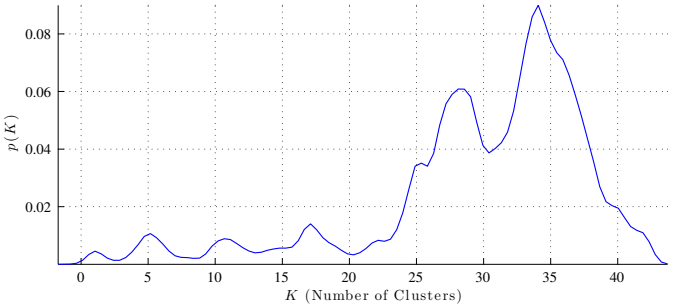
Among the alternative priors, the "Prop." prior was the top performer. Its overall performance was only bested by the small fixed α variant of the DP prior. This is because the prior actually expresses a strong preference for a small number of clusters. Unfortunately, there is no principled way to calibrate the strength of this preference since we cannot link the parameters of the prior to the expected number of clusters. None of the alternative priors produced a MAP estimate of Z that preserved the true track nor did any have a significant mode near $K = 1$ in the resulting distribution of the number of clusters. Even the MAP estimate from "Prop." prior split the true track into two overlapping fragments. However, the "Prop." prior still managed to convey a decent picture of the situation through its weighted-edge-sample plot. The plots from all five of the alternative priors can be seen in figure 5.16.

We turn now to the question of whether or not each move should consider preserving the status quo. Table D.3 reveals that modifying the proposal distribution, so that each move includes the current state among the feasible candidate states, can universally be regarded here as beneficial in terms of the scores (both individual and combined) and the accuracy of the estimates, which have improved considerably in most cases. Using this configuration of the sampler, all of the nonparametric priors with a small fixed α produced estimates that were nearly perfect in their accuracy. For comparison with the results from the default implementation of the proposal distribution, see figures 5.17 and 5.18 for the weighted-edge-sample plots from the scan-constrained nonparametric priors and the alternative priors, respectively.

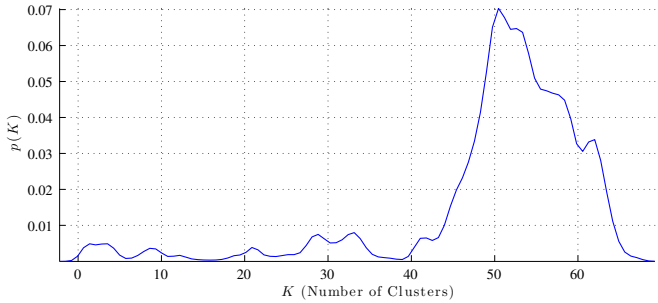
Regarding the results from this alternative implementation of the proposal distribution, the only thing that gives us pause is that for every experiment the MAP probabilities are lower than what was obtained using the default implementation. Thus, the sampler is clearly failing to explore some regions of high probability and the convergence of the Monte Carlo estimates (i.e. the ergodic averages) is doubtful. However, the chain has spent its allotted time exploring meaningful parts of the space and produced results that agree well with ground truth. Did we just get lucky or have we managed to bias the search in a productive manner through our choice of proposal distribution?



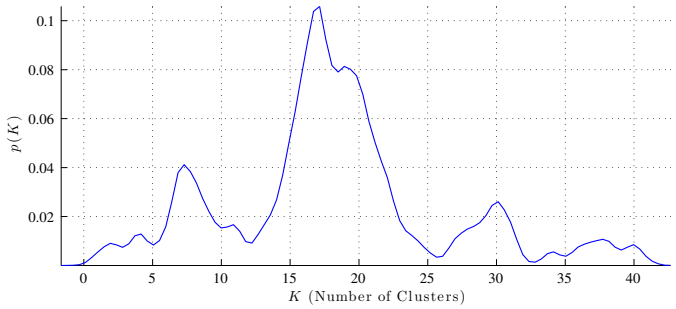
(a) Scan-constrained DP prior.



(b) Scan-constrained UP prior.

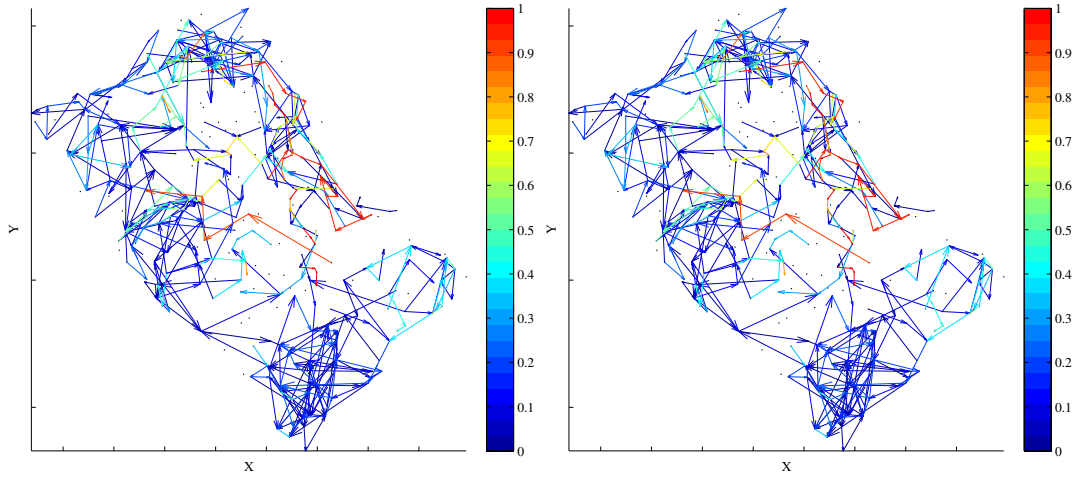


(c) Scan-constrained EP prior.



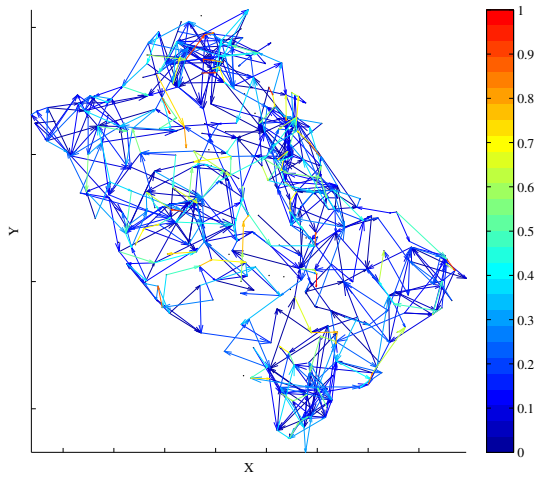
(d) Scan-constrained EP2 prior.

Figure 5.14: Cluster count distributions from the scan-constrained nonparametric priors. The results were derived from an MCMC sample of the one target data set.



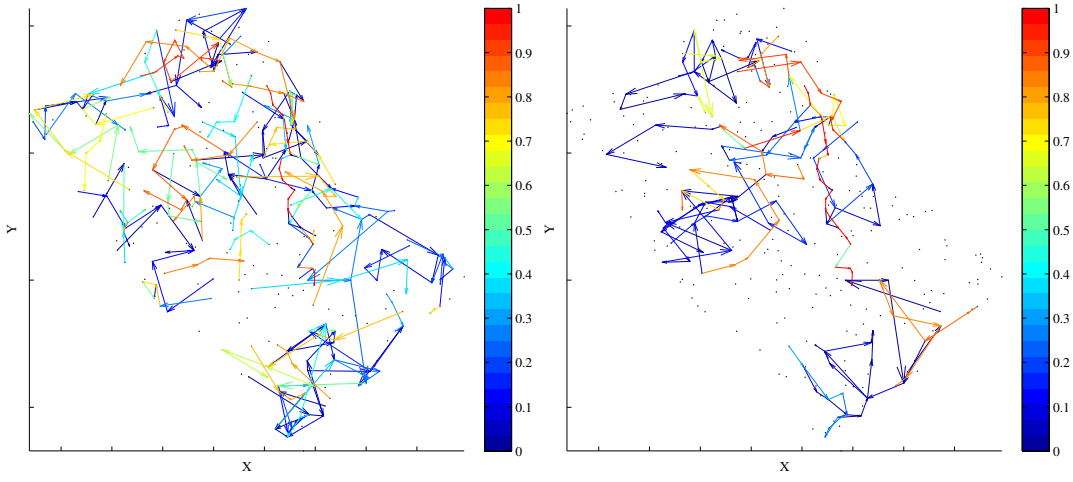
(a) Scan-constrained UP prior.

(b) Scan-constrained EP prior.



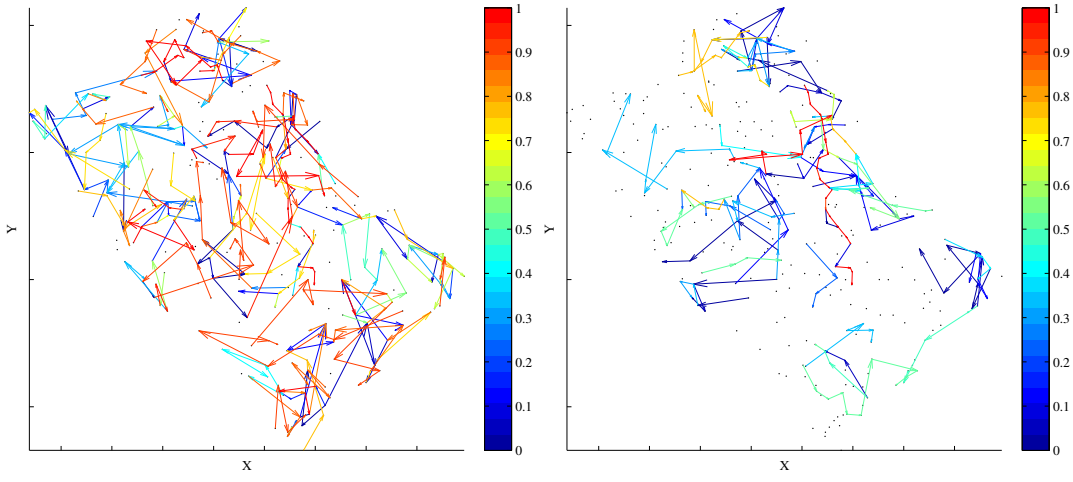
(c) Scan-constrained EP2 prior.

Figure 5.15: Output from the scan-constrained nonparametric priors not previously shown. The results were derived from an MCMC sample of the one target data set.



(a) HUP prior.

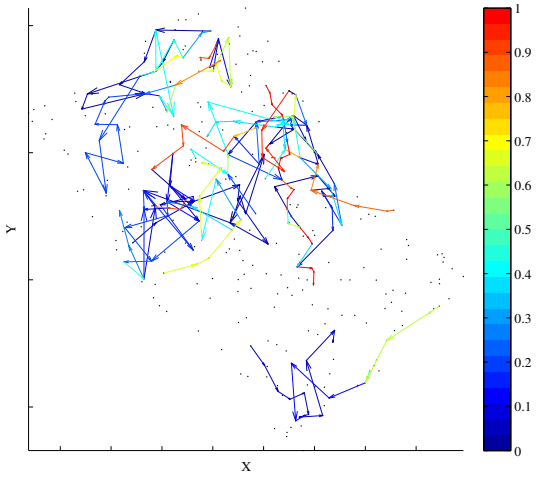
(b) Scan-constrained HUP prior.



(c) "K" prior.

(d) "Prop." prior.

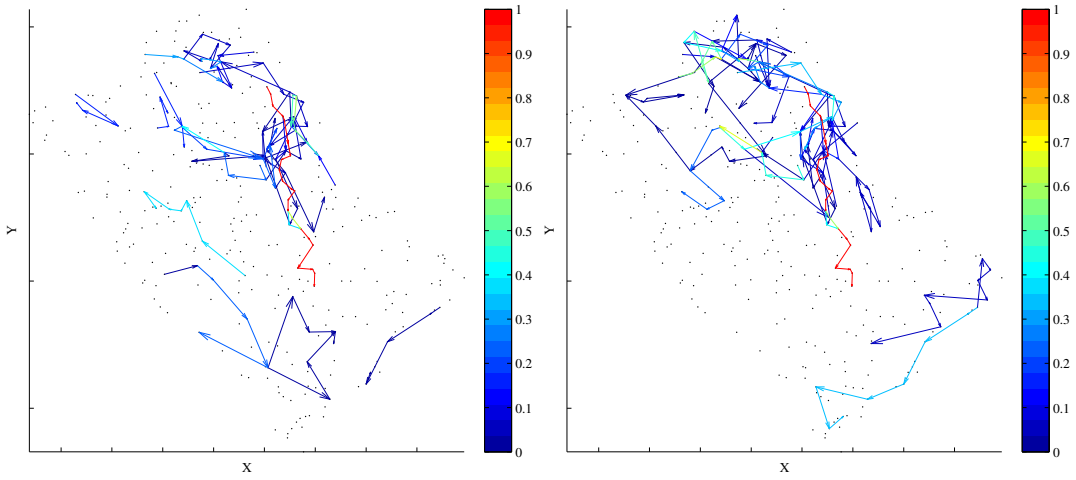
Figure 5.16: Output from the alternative priors. The results were derived from an MCMC sample of the one target data set.



(e) “Oh” prior.

Continuation of Figure 5.16: Output from the alternative priors. The results were derived from an MCMC sample of the one target data set.

One concern was that the sampler became trapped around a local optimum. However, the usual diagnostics did not reveal any significant differences between the two proposal distributions. Trace plots showed relatively decent movement in the statistics, similar to what we observed with the default proposal distribution. The acceptance rate appeared to be reasonable as well. Using the default proposal distribution, the average acceptance rate was 0.30, with some minor variation across experiments. The alternative implementation of the proposal distribution led to a modest drop; down to 0.26, on average. Unfortunately, the optimal rate for this type of problem is not known and we are not aware of any published results for similar problems. In terms of the autocorrelation within the sample, there was no significant change between the two proposal distributions. In both cases, it was poor compared to the ideal, decaying in a roughly linear fashion with decorrelation around 800. We also tried increasing the number of sampling iterations to 25,000 in order to see if the sampler might manage explore more of the space given additional time, but the results remained comparable. The lack of convergence will likely have the most impact on our variance estimates and the quality of the very uncertainty that we were motivated to quantify in the first place. Of course, the precision of these estimates was never expected to be high because of the inherent complexity of the problem and the resource constraints. Our goal from the outset was to do the best we could under the circumstances and offer at least a rough



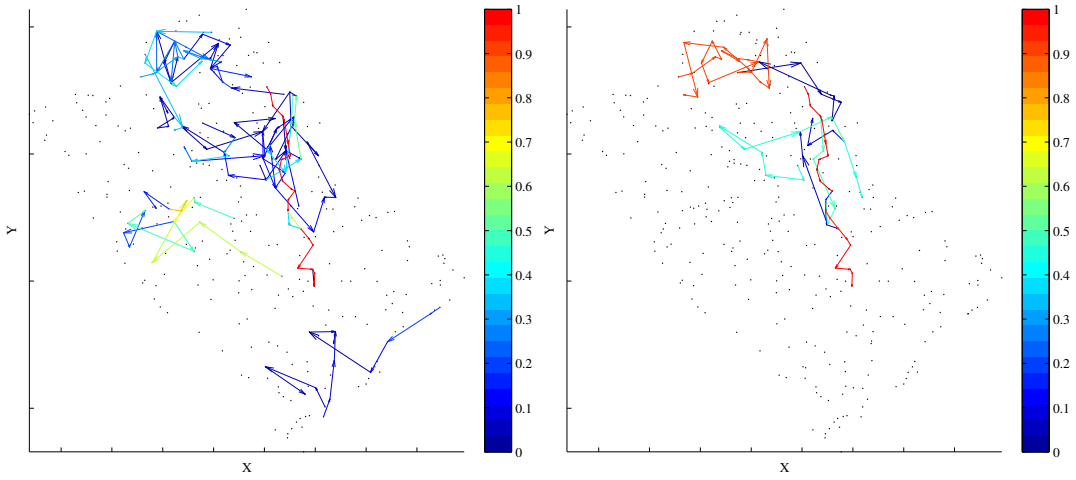
(a) Scan-constrained DP prior.

(b) Scan-constrained UP prior.

(c) Scan-constrained EP prior.

(d) Scan-constrained EP2 prior.

Figure 5.17: Output from the scan-constrained nonparametric priors when the individual moves in the proposal distribution were allowed to propose remaining in the current state. The results were derived from an MCMC sample of the one target data set.



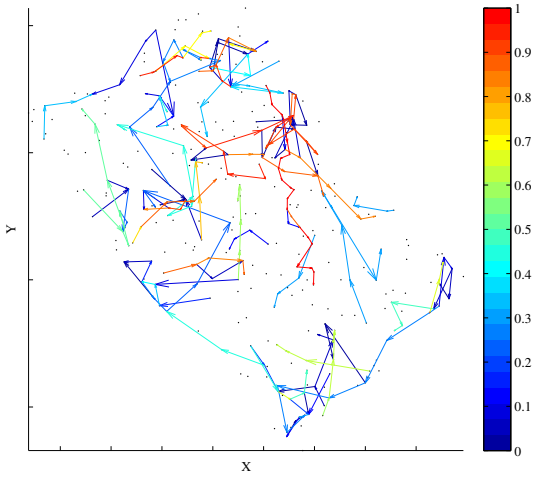
(a) HUP prior.

(b) Scan-constrained HUP prior.

(c) "K" prior.

(d) "Prop." prior.

Figure 5.18: Output from the alternative priors when the individual moves in the proposal distribution were allowed to propose remaining in the current state. The results were derived from an MCMC sample of the one target data set.

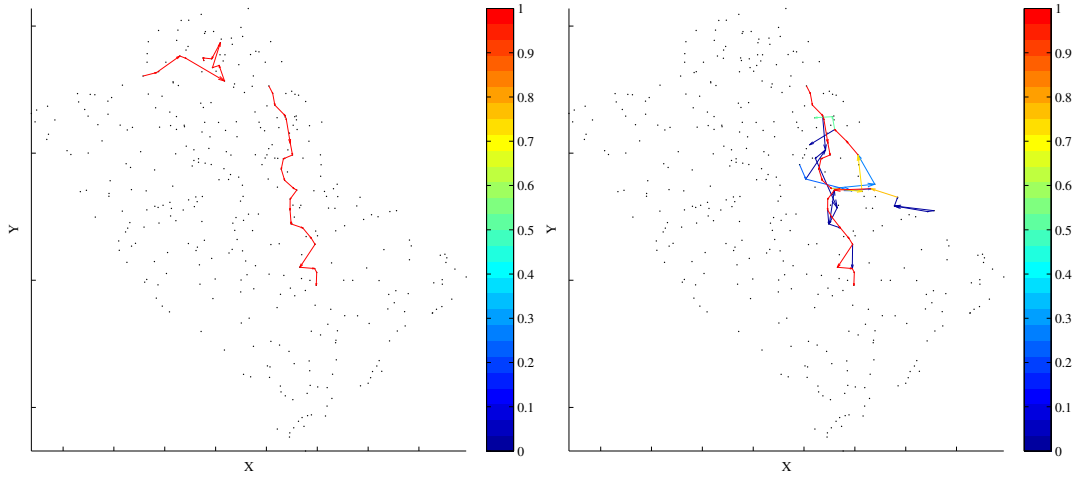


(e) "Prop." prior.

Continuation of Figure 5.18: Output from the alternative priors when the individual moves in the proposal distirbution were allowed to propose remaining in the current state. The results were derived from an MCMC sample of the one target data set.

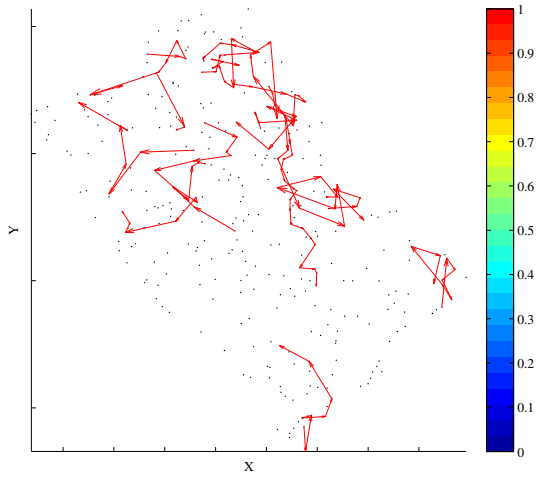
estimate to fill the present void. It is left to future work to improve those estimates.

Because of our concerns about the convergence of the estimates computed from the MCMC sample, we investigated PMC sampling as a potential remedy. Using the default proposal distribution, we can see from table D.5 that the scores and estimates are generally better than their MCMC-derived counterparts while the relative ranking of the priors remained largely unchanged. However, one glaring exception to this trend toward the better are the maximum-score estimates, which are no longer exemplary. This is because, as we noted earlier, the highest scoring configurations are not among the most likely according to the model. Thus, the PMC sample does not include elements representing those configurations. As a result, the respective maximum-score and MAP estimates are much closer. To visualize the spectrum of results from the PMC samples using the default proposal distribution, figure 5.19 shows the weighted-edge-sample plots from a selection of priors. The selection includes the DP prior with a small fixed α, which had the highest expected combined score, the EP prior with a small fixed α, which represents a middling combined score, and the "K" prior, which had the lowest expected combined score. Note that despite the fact that the second panel in the figure looks like an accurate picture of what was going on, the individual track score reveals that the target is being split into two adjacent tracks.



(a) DP prior with $\alpha = 1 \times 10^{-6}$.

(b) EP prior with $\alpha = 1 \times 10^{-6}$.



(c) "K" prior.

Figure 5.19: Example of the output from the highest (first panel), middling (second panel), and lowest scoring (last panel) PMC samples of the one target data set.

The other glaring exception to the improvement trend are the variance estimates. While the number of targets estimated from the PMC sample was more accurate in almost all of the experiments, the variance estimates have all gone to zero, which is entirely unconvincing. Note that the approximate 99% confidence intervals for the corresponding mean estimates were all confirmed to be reasonably small. Such estimates convey no uncertainty and thus are poorly suited to surveillance applications. A typical result obtained from the PMC sampler can be characterized as offering a good depiction of the true target plus erroneous targets, all with equal probability. These results are symptomatic of weight collapse (i.e. degeneracy), which is when one element comes to dominate the sample. This indeed turned out to be a major problem. Table 5.2 shows the average weight variances and effective sample sizes (ESS) for all of the PMC experiments. The ESS was computed for the sample at each iteration and the values in the table are the averages over all iterations. Clearly, those are not the kind of numbers that we had hoped to see. Although they did tend to decrease slightly with each PMC iteration, the variances remained just below the maximum value of 1×10^{-3} (i.e. one sample element has all of the weight). All of the ESS values were disappointingly small. Despite our hope that using an initial proposal distribution capable of generating widely varying cluster assignments would encourage a diverse sample, there was very little diversity within the final sample. Note that the problem does not appear to be overly sensitive to the initial proposal distribution because our experiments with the original implementation discussed in section 2.4.3.4 led to similar results. There is a moderate positive correlation ($r = 0.63$) between the expected score, which we can regard as a proxy for the accuracy of the estimates, and the average ESS. Since the accuracy is highly dependent on the choice of prior, this appears to affirm the importance of that decision.

When using the alternative implementation of the proposal distribution with the PMC sampler, we again saw improvements in the scores (both individual and combined) and the accuracy of the estimates. Compared to the results of the MCMC experiments, the improvement over the default proposal distribution was modest here as we might expect given the better baseline performance of the PMC sampler. Just as we saw with the MCMC sampler, switching to the alternative implementation of the proposal distribution led to all of the nonparametric priors with a small fixed α producing estimates that were nearly perfect in their accuracy. This agreement is encouraging

Prior	Avg. Variance			Avg. ESS		
	Default	Status Quo	K-add	Default	Status Quo	K-add
DP	6.21e-04	3.89e-04	7.16e-04	2.4	3.9	1.8
DP (constr.)	5.96e-04	5.24e-04	8.06e-04	2.2	5.6	1.3
DP (fixed α)	6.32e-04	3.19e-06	8.95e-05	16.4	637.3	370.8
UP	7.18e-04	8.04e-04	7.32e-04	1.5	1.4	1.6
UP (constr.)	6.43e-04	6.27e-04	9.06e-04	2.0	3.5	1.1
UP (fixed α)	2.81e-04	6.23e-05	6.67e-04	7.3	224.5	2.2
EP	8.26e-04	5.60e-04	6.66e-04	1.2	2.2	1.8
EP (constr.)	8.67e-04	6.76e-04	7.07e-04	1.2	3.6	1.7
EP (fixed α)	3.24e-04	1.33e-04	9.08e-06	44.8	426.1	487.4
EP2	4.92e-04	4.79e-04	9.87e-04	2.6	4.5	1.0
EP2 (constr.)	5.46e-04	3.32e-04	8.81e-04	2.5	9.4	1.3
EP2 (fixed α)	4.85e-04	1.33e-04	1.06e-05	5.5	336.8	544.8
HUP	6.58e-04	7.83e-04	9.61e-04	3.6	6.4	1.0
HUP (constr.)	6.16e-04	1.45e-04	5.04e-06	7.2	282.8	626.7
K	8.56e-04	4.09e-04	7.96e-04	1.2	3.2	1.5
Prop.	6.98e-04	5.78e-04	3.31e-05	3.4	14.2	613.3
Oh	5.93e-04	4.26e-04	1.50e-05	2.0	3.0	607.9

Table 5.2: Average effective sample sizes for the results from the PMC sampler on the one target data set.

since it suggests that, despite our concerns about the mixing of the chain, the MCMC estimates based on the alternative proposal distribution are not entirely unreasonable.

Referring again to table 5.2, we can see that the problem of weight degeneracy, while still with us, was a bit less severe in all but one experiment when using the alternative implementation of the proposal distribution with the PMC sampler. The effective sample sizes for the nonparametric priors with a small fixed α and the scan-constrained HUP prior have improved considerably, although all but one are still less than half of the actual sample size. Nonetheless, these results do suggest that modifying the proposal distribution is a promising way forward. One could argue that the alternative implementation of the proposal distribution generally performed better because it is closer to the posterior distribution that it is trying to represent. However, we are at a loss to explain the stark contrast between the performance given different choices for the prior over cluster labels since the influence of the exchangeable (and approximately exchangeable) priors on the proposal is computed in the same manner, independent of the prior. Note that only the scan-constrained HUP and "Oh" priors do not inform the proposal distribution.

Finally, let us look at the performance of the PMC sampler using the k-add mixture proposal distribution. This implementation of the proposal distribution was

intended to test the concept of completely relying on the add and delete moves, and the results were decidedly mixed. Tables D.9 and D.10 show that the scores (again; both individual and combined) and estimates are by and large better than anything else that we evaluated. The tendency to overestimate the number of targets was reduced considerably using this configuration, with the expected number being one or two in more than half of the experiments and only exceeding three in three experiments. This led to generally higher expected combined scores than the other configurations. Looking at the individual scores, we can see that the target itself was almost completely identified in every experiment, where the combined scores reflect the number of spurious associations. Despite the improved scores, the problem of weight collapse remained and varied even more with the choice of prior. To our surprise, both the "Prop." and "Oh" priors led to decent ESS's and estimates that were nearly perfect in their accuracy. It is not clear why the change in the proposal distribution effected these two priors so dramatically.

The k-add mixture proposal distribution also served as an example of an independence sampler because each move was made independently of the current state. Because of the way that we defined the moves, the k-add mixture was the only proposal distribution, among the three evaluated here, capable of transitioning from any number of clusters to any other number of clusters (subject to a fixed maximum value). This freedom of movement was intended to prevent the sampler from getting trapped by failing to move through low probability regions. Therefore, the results obtained using the k-add mixture proposal can serve as a check against the others. They do not in fact reveal any high probability regions that were missed by the other configurations. On the contrary, despite the extra freedom of movement, the MAP probabilities from these samples were generally lower than what we saw from the other proposal distributions, but close (0.6%, on average) to what we saw when using the PMC sampler with the alternative implementation of the proposal distribution. This does, however, give us some confidence that those PMC experiments were not merely the outcome of a fortuitous run.

5.5.5 Five target

This data set consists of five targets walking in mountain terrain approximately 5 km north of the sensor. It was the same one used earlier for the feasible graph evalu-

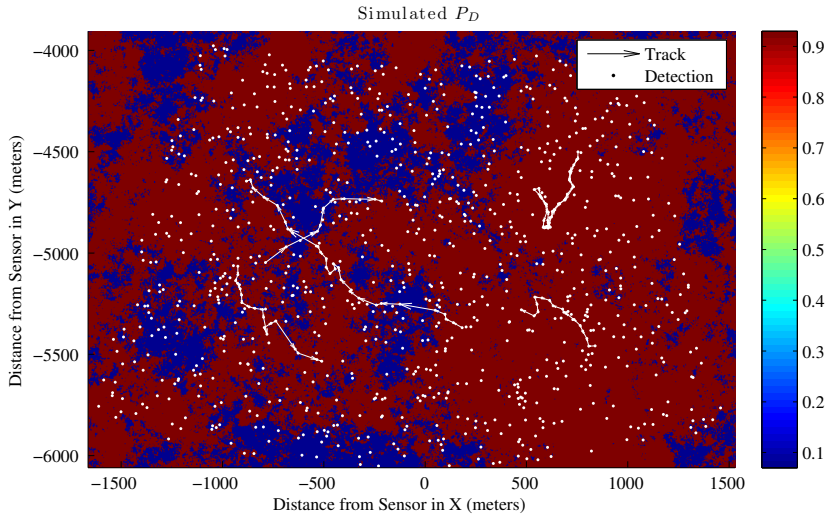


Figure 5.20: Simulated P_D with the detections and tracks for the five target data set.

ation with $T = 30\,\mathrm{s}$ and the randomly generated binary P_D model. Figure 5.20 shows the P_D along with the detections and true tracks. Notice how the two targets on the right-hand side of the figure are moving in regions of high P_D. The other three targets pass through numerous regions of low P_D.

To remain consistent with previous experiments, the feasible graph was constructed using the PSH/C4 model with the maximum out-degree of each vertex limited to 5. This yielded complete coverage of the degree-one edges. For the three targets on the left-hand side of the figure, that coverage was achieved despite many missed detections. This sparseness, however, led to lower weights for the degree-one edges and poor coverage for the edges of higher degree which makes the tracker's job more difficult. Indeed, as we will see, the results are not as good for track 5, which ends near $(-214, -4744)$ and consists of just nine measurements, and track 3, which ends near $(-882, -4646)$.

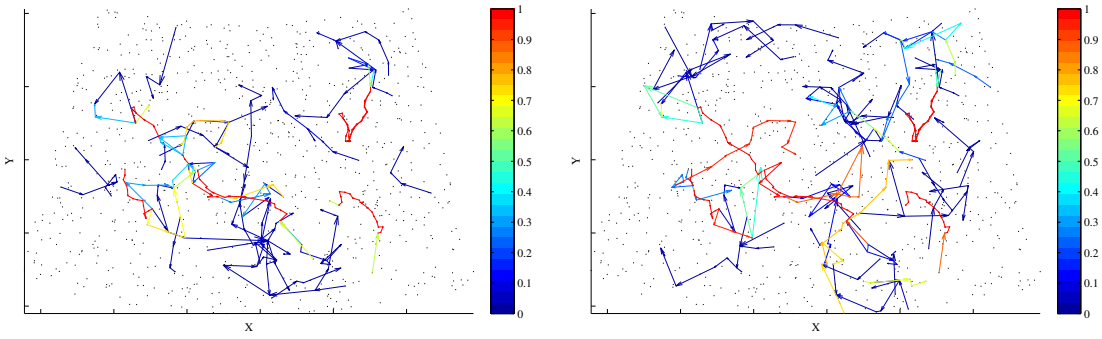
Tables D.11–D.20 contain the summary statistics from the experiments.

Looking first at the performance of the MCMC sampler, we can once again see that when using the default implementation of the proposal distribution, there was a tendency to grossly overestimate the number of targets present. As we would expect given the fivefold increase in the number of true targets, the estimates of K are larger than what we saw using the one target data set and contain more variance. The DP prior

using a small fixed α again produced the most reasonable estimate, 5.2 ± 0.75, with the "K" prior way off the mark at 102.2 ± 28.49. The tendency to overestimate the number of targets primarily reflects the accuracy of the model, which in turn is largely determined by the choice of the prior distribution over the cluster labels. As evidenced by the large gap between the MAP estimate and true value of K, most choices of the prior resulted in inaccurate posterior models. The estimates based on the nonparametric priors with a small fixed α were, however, generally reasonable and confirm the notion that a strong preference for the lowest possible number of clusters is desirable for this application. Of course, when the likelihood component of the model is dominant then an uninformative prior such as the scan-constrained HUP can also be effective. That appears to have been the case here as it too produced a reasonable estimate of 6.1 ± 1.86, although we must bear in mind that this could be an artifact of the approximations employed by the scan-constrained HUP prior; additional experiments would need to confirm this. The weighted-edge-sample plot for the weakly informative UP prior with a small fixed α does paint a pretty good picture of what was happening in the surveillance region which can be seen, along with the results from the other nonparametric priors, in figure 5.21. This lends some support to the dominance of the likelihood component here. The results from the alternative priors can be seen in figure 5.22.

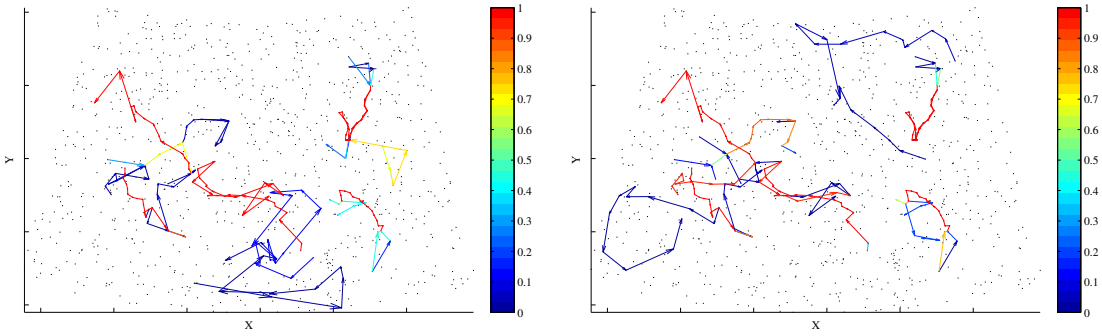
Just as we saw using the one target data set, introducing the scan constraints into the concentration parameter hyperprior produced lower expected α values yet the estimates remained well above their ideal values. This supports our earlier conclusion that the definition of the hyperprior should be revisited with an eye toward strengthening the preference for small values. More will be said about how that could be done when we discuss future work. The false alarm rate was again underestimated to varying degrees as we would expected given the overestimates of the number of targets. The MAP estimates of λ, however, were generally better than those obtained from the previous experiments. This is likely due to the method used to set the hyperparameters, which resulted here in a hyperprior for the false alarm rate that favored larger values than the actual rate.

Using the alternative implementation of the proposal distribution, which allowed each move to consider preserving the status quo, was again largely beneficial in terms of both the scores and estimates. However, here it was the combined scores that generally improved while the individual scores dropped slightly in most cases. We can



(a) DP prior with $\alpha = 1 \times 10^{-6}$.

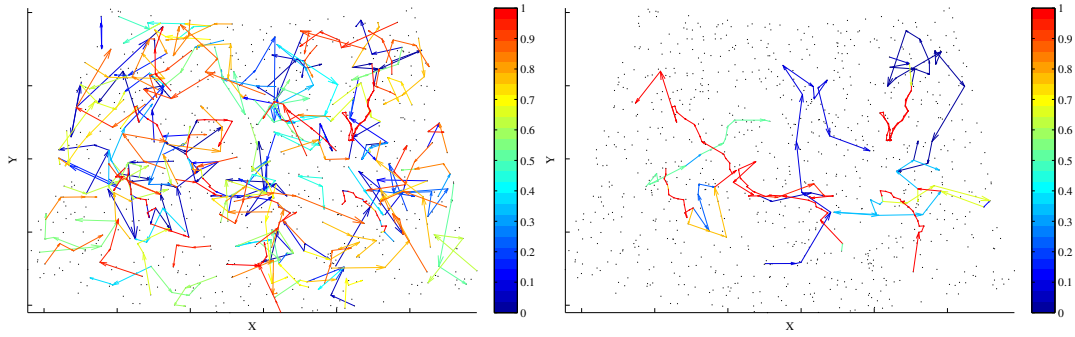
(b) UP prior with $\alpha = 1 \times 10^{-6}$.



(c) EP prior with $\alpha = 1 \times 10^{-6}$.

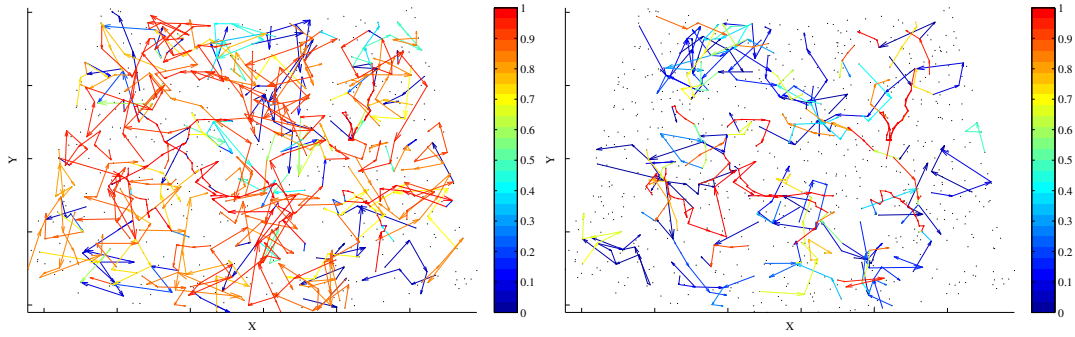
(d) EP2 prior with $\alpha = 1 \times 10^{-6}$.

Figure 5.21: Output from the nonparametric priors. The results were derived from an MCMC sample of the five target data set.



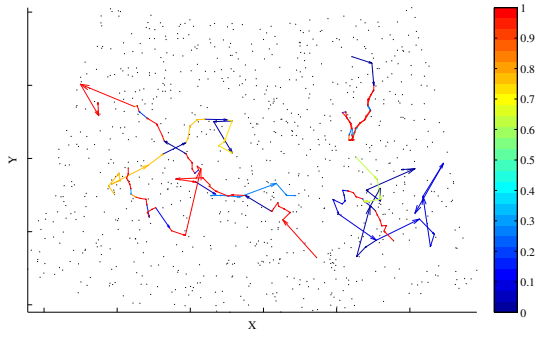
(a) HUP prior.

(b) Scan-constrained HUP prior.



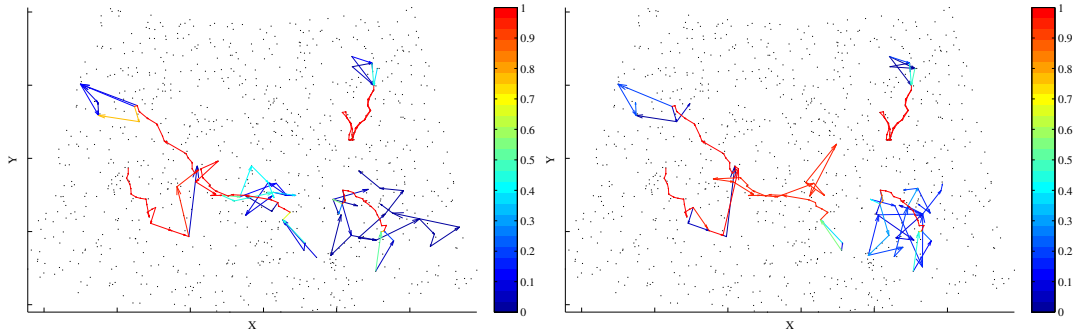
(c) "K" prior.

(d) "Prop." prior.



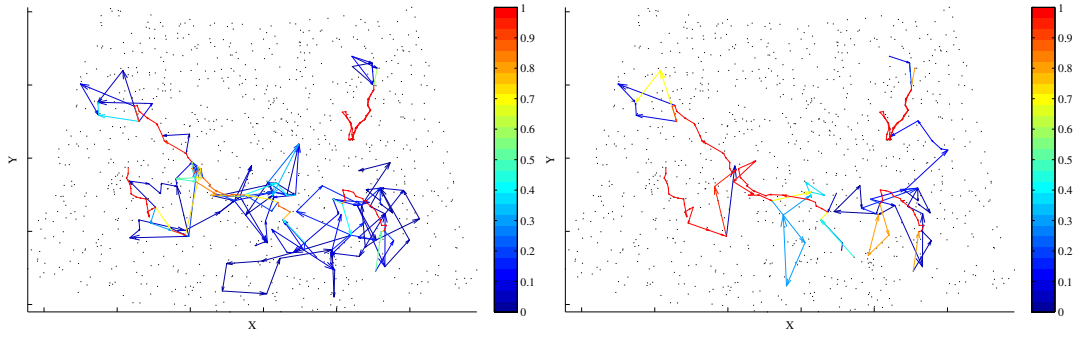
(e) "Oh" prior.

Figure 5.22: Output from the alternative priors. The results were derived from an MCMC sample of the five target data set.



(a) Scan-constrained DP prior.

(b) Scan-constrained UP prior.



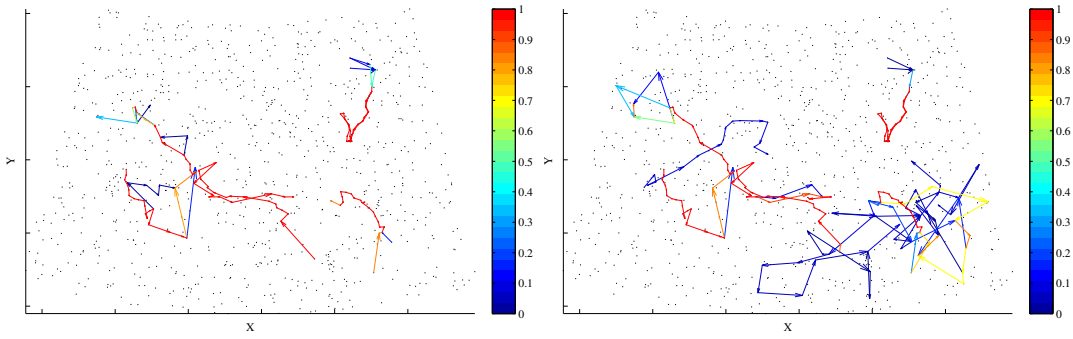
(c) Scan-constrained EP prior.

(d) Scan-constrained EP2 prior.

Figure 5.23: Output from the scan-constrained nonparametric priors when the individual moves in the proposal distribution were allowed to propose remaining in the current state. The results were derived from an MCMC sample of the five target data set.

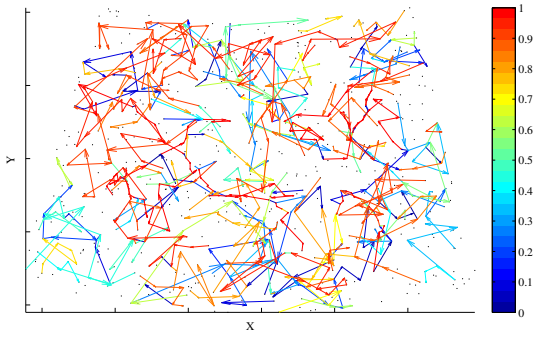
attribute the improvement in the combined scores to the fact that by and large the number of targets was no longer overestimated. As table D.13 indicates, there was actually a tendency to slightly underestimate K. Only the three property-based priors led to overestimates, while in the majority of the experiments the actual number of targets was underestimated by just one. In table D.14, we can see that this is because the weakly resolved target 5 was frequently missed; sometimes completely. The estimates of α and λ improved considerably. Compared to default implementation of the proposal distribution, the acceptance rate dropped slightly from 0.30 to 0.28. To visualize the results from these experiments, see figures 5.23 and 5.24 for the weighted-edge-sample plots from the scan-constrained nonparametric priors and the alternative priors, respectively.

Despite the trend of improved summary statistics, we remained concerned that

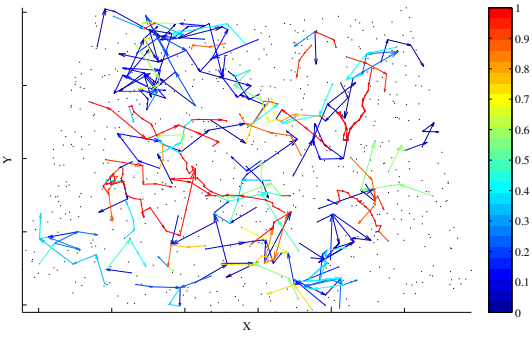


(a) HUP prior.

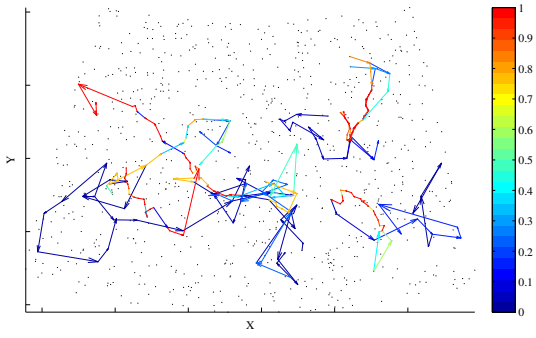
(b) Scan-constrained HUP prior.



(c) "K" prior.



(d) "Prop." prior.



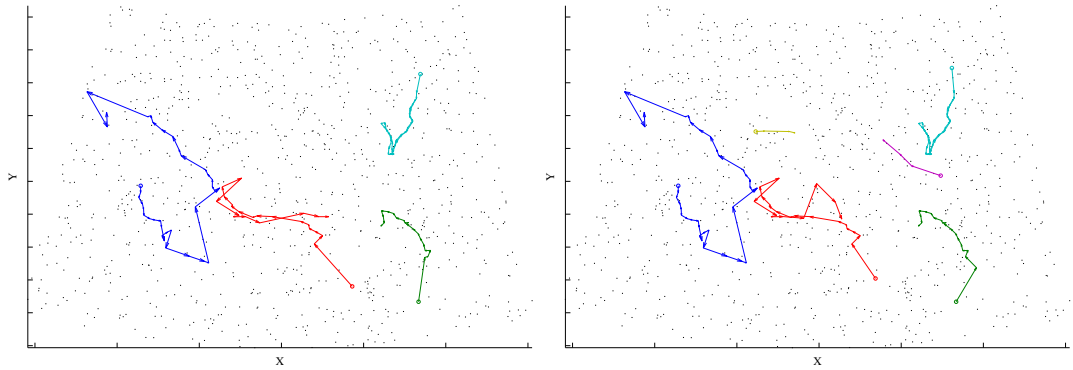
(e) "Oh" prior.

Figure 5.24: Output from the alternative priors when the individual moves in the proposal distribution were allowed to propose remaining in the current state. The results were derived from an MCMC sample of the five target data set.

the alternative implementation of the proposal distribution was causing the sampler to linger around a local optimum. In particular, we suspected that by setting the initial state to all false alarms we were forcing the sampler to approach, in terms of the number of targets, any modes from below where it would run the risk of getting trapped around the dominant mode with the lowest number of targets. To test this theory, we re-ran the three experiments corresponding to the DP prior (unconstrained, scan-constrained, and with a small fixed α) with the initial state set to 10 targets instead of zero. The initialization was implemented using ten consecutive add moves that did not consider keeping the status quo.

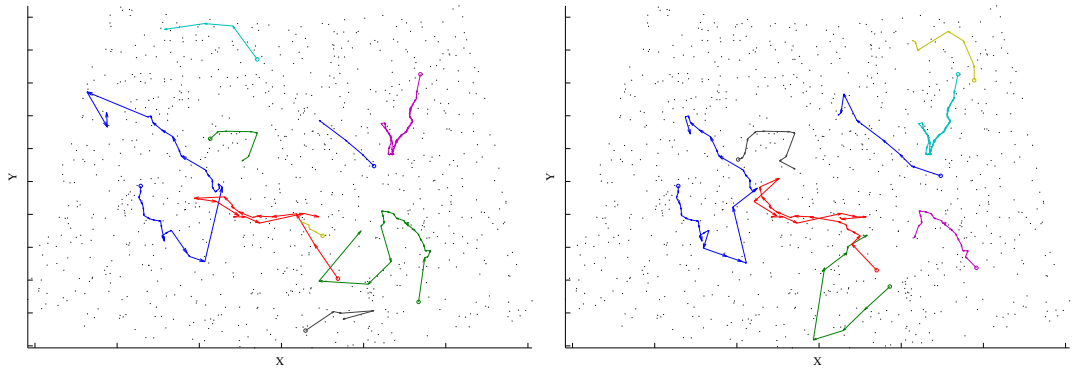
The results from these new experiments were mixed. Only the DP prior with a small fixed α produced an estimate that was close to the new initial value, while the others produced estimates near those obtained using the original zero target initialization. The expected number of targets for the unconstrained, scan-constrained, and small fixed α DP priors were 4.41 ± 1.27, 5.05 ± 1.22, and 9.00 ± 0.06, respectively. Although the first two exhibited some desirable insensitivity to the initial conditions, they still failed to reach the higher probability configurations that were visited when using the default proposal distribution. The results suggest that there is a local mode near the true configuration. Unfortunately, the results also suggest that there is indeed some attraction to local modes due to the probability of remaining in the current state dominating the choices in the add move. In the third case, the strong prior beliefs appear to have exacerbated the problem, causing the sampler to get stuck near the initial configuration. To see if this problem is related to the particular preferences expressed by the DP, we repeated the ten target initialization experiments for the UP and EP2 priors, both with a small fixed α. We also evaluated the scan-constrained HUP prior using the ten target initialization for comparison. The expected K values were 8.00 ± 0.13, 7.01 ± 0.18, and 5.6 ± 0.87, respectively. These results suggest that strong prior beliefs of any kind can increase the likelihood that the sampler gets stuck when each move considers the current state among the candidates. The uninformative scan-constrained HUP prior led to similar results for both proposal distributions and both initialization methods. Figure 5.25 shows the MAP estimates of Z from the ten target initialization experiments for comparison.

Together with the chosen initialization scheme, the alternative implementation of the proposal distribution has effectively biased the search in favor of configurations



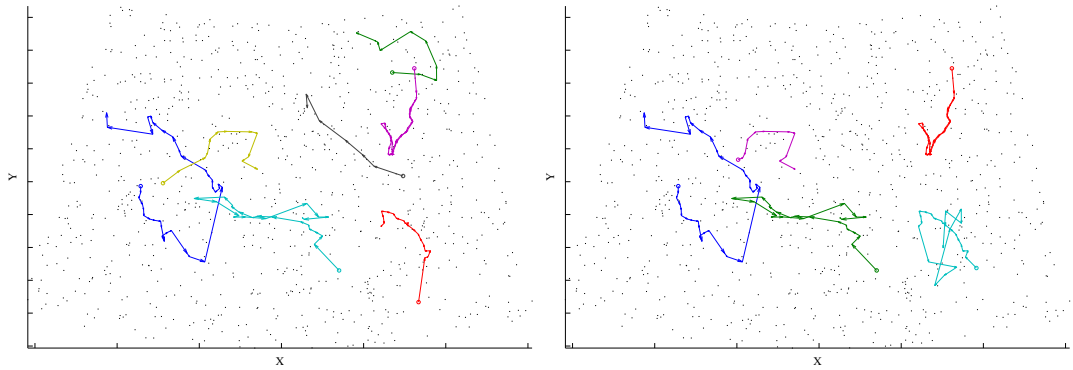
(a) Unconstrained DP prior.

(b) Scan-constrained DP prior.



(c) DP prior with $\alpha = 1 \times 10^{-6}$.

(d) UP prior with $\alpha = 1 \times 10^{-6}$.



(e) EP2 prior with $\alpha = 1 \times 10^{-6}$.

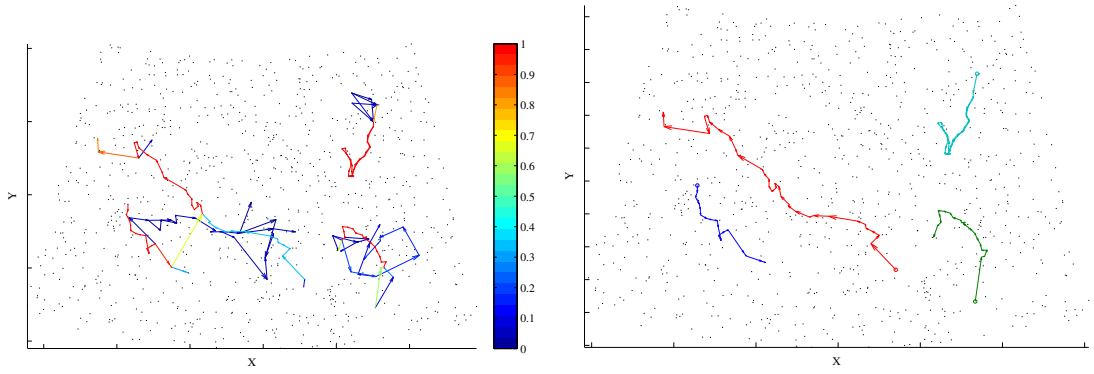
(f) Scan-constrained HUP prior.

Figure 5.25: MAP estimates of Z from the experiments using a ten target initialization when the individual moves in the proposal distribution were allowed to propose remaining in the current state. The results were derived from an MCMC sample of the five target data set.

with fewer targets. Although problematic in terms of its implications for how efficiently the sampler is exploring the space, this bias was not particularly harmful here, or in the experiments using the one target data set, because we actually have an a priori preference for such configurations. In this case, it actually masked some of the deficiencies with our choice of cluster label prior, hence our model, and provided a measure of fail-safe behavior. However, we should not count on always having such good fortune. Instead of preventing a gross overestimate, getting trapped near the first dominant mode may very well lead to a gross underestimate. These results suggest that in addition to ensuring that the proposal distribution can move the sampler away from the current state with sufficient probability, care should be taken to validate the model independently of the sampler. One approach would be to use stochastic optimization techniques to derive a MAP estimate that can be compared to the output of the sampler or ground truth, if available.

By allowing the current state to compete with the other feasible states when generating each move, the alternative implementation has also effectively reduced the "scale" of the proposal distribution by taking away probability mass from those options that lead to deviations from the status quo. In theory, including the current state among the candidates for each move does not present a problem since it lies within the support of optimal proposal distribution. However, in practice we can only approximate the optimal proposal distribution and there is a risk that the direct estimate of the probability of the current state will be too large compared to averaged estimates for the other states used by many of the moves. The scale of the proposal distribution is critical to the mixing speed of the chain. When the scale is too small, the chain will move too slowly; too large and most of the proposals will be rejected.

Since the indicator variables in $(Z, Z0)$ are discrete-valued and have well-defined structure, the scale of our proposal distribution is not determined by a scalar standard deviation parameter, which is the typical situation, but rather through the choice and implementation of the moves in the mixture. Our motivation was to create a set of moves that could make large and meaningful changes to the state in order to get reasonable estimates with a limited sample size. The results from these experiments suggest that we have only been partially successful in that endeavor. Future work should explore ways to increase the scale, by modifying the moves to propose larger changes to the current state, while maintaining a reasonable acceptance rate. With that said, we



(a) Weighted edge samples.

(b) MAP estimate of Z.

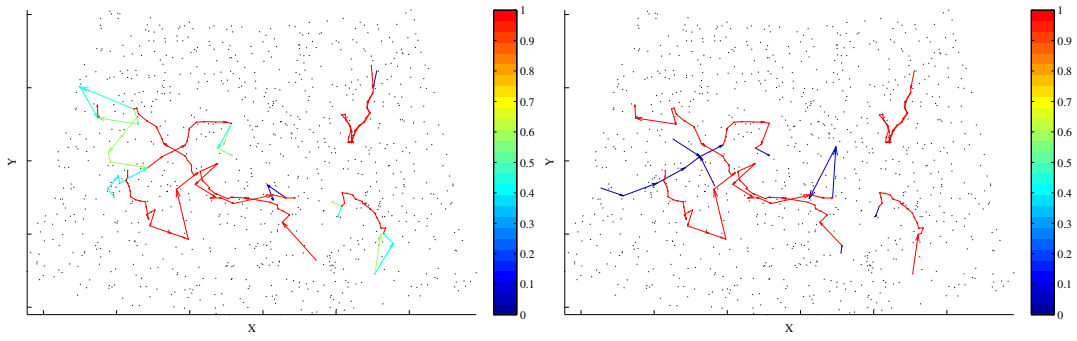
Figure 5.26: Output from the scan-constrained DP prior when all moves in the proposal distribution were allowed to propose remaining in the current state and the probability of each selection was computed using an estimate of the posterior. The results were derived from an MCMC sample of the five target data set.

would like to get a sense of what happens when we minimize the scale of the currently defined proposal distribution as a baseline. To do this, we not only allowed each move to propose the current state but forced it to make all selections based on the posterior approximation. In other words, we undid the modifications to the original add and delete move definitions that were proposed earlier. Figure 5.26 shows the output from the scan-constrained DP prior, which we have taken to be the canonical example. The results are reasonable albeit incomplete. The weakly resolved target 5 is missing but the MAP estimate of Z is otherwise a good one; with a combined score of 0.80 and individual target scores of: 1.0, 0.99, 0.92, 0.87, and 0.0, respectively. Because of the small number of measurements and the relatively low likelihood for target 5, potential candidate states containing that track had a small probability of being selected; too small for the given sample size. However, this is primarily a reflection of the model rather than a deficiency with the posterior approximation used to weight the selections in each move since the posterior probability of the true state is actually much lower than the MAP estimate. It does appear that we can claim some measure of success in facilitating meaningful moves.

Turning now to the experiments based on the PMC sampler, we can see from table D.15 that the scores and estimates were generally better than their MCMC-derived counterparts, as was the case for the one target data set. However, the there were also

some notable exceptions to this trend. The performance of the DP, EP, and EP2 non-parametric priors using a small fixed α, which was previously very good, worsened with PMC sampling. The performance of the previously successful scan-constrained HUP prior also worsened, although this could be a consequence of the approximation scheme used to implement it. As we saw in the one target experiments, switching the sampling algorithm caused the maximum-score estimates to drop toward the MAP values. However, this time the decline was less severe, with almost all the scores remaining above 0.50. Surprisingly, the "K" prior was among those with a MAP score of more than 0.50. The variance notwithstanding, it even managed to produce a reasonable estimate of the number of clusters: 7.0 ± 0.0. The "Prop." and "Oh" priors, which performed very poorly with the MCMC sampler, also fell into this category, with the "Oh" prior being one of just three configurations to produce the correct estimate of the number of clusters. To visualize the spectrum of results from the PMC samples using the default proposal distribution, figure 5.27 shows the weighted-edge-sample plots from the UP and HUP priors, which were tied for the highest expected combined score, the DP prior with a small fixed α, which represents a middling combined score, and the scan-constrained DP prior, which had the lowest expected combined score. Notice that the challenging target 5 was partially identified in both the high and lowing scoring configurations, but not the middling. The lowest scoring configuration actually did a reasonable job identifying each of the targets; it just contained the most spurious associations.

Unfortunately, the variance estimates were still anomalously small and the problem of weight collapse remained acute. Table 5.3 shows the average weight variances and effective sample sizes (ESS) for all of the PMC experiments, computed as described in the previous section. Regarding the weight variances, we can see that they were less than the maximum value of 1×10^{-3}, but not by much. However, they were marginally better overall than what we saw for the one target data set. An examination of the variance trend for each experiment confirmed that the variance of the weights tended to decrease modestly over the PMC iterations. Regarding the ESS's, almost all were unacceptably small. Only the "K" prior managed to produce a reasonable sample according to this metric, however, it is misleading. What we are really seeing is an example of how the ESS is an imperfect diagnostic tool. In this case, the sample elements became essentially equally weighted, which led to a perfect ESS of 1000 and a high average over all iterations, yet the sample is not representative of the distribution.



(a) Unconstrained UP prior.

(b) HUP prior.

(c) DP prior with $\alpha = 1 \times 10^{-6}$.

(d) Scan-constrained DP prior.

Figure 5.27: Example of the output from the highest (first two panels), middling (third panel), and lowest scoring (last panel) PMC samples of the five target data set.

Prior	Avg. Variance			Avg. ESS		
	Default	Status Quo	K-add	Default	Status Quo	K-add
DP	5.75e-04	8.26e-04	8.97e-04	2.8	1.4	1.2
DP (constr.)	4.90e-04	1.55e-04	9.16e-04	2.6	87.6	1.2
DP (fixed α)	3.68e-04	1.43e-04	9.13e-04	7.4	686.9	1.1
UP	5.98e-04	1.41e-04	9.78e-04	2.6	280.9	1.0
UP (constr.)	4.45e-04	3.73e-04	6.94e-04	6.6	88.6	1.7
UP (fixed α)	3.71e-04	7.43e-04	9.86e-04	7.6	3.9	1.0
EP	1.77e-04	2.83e-04	8.99e-04	21.5	19.1	1.2
EP (constr.)	1.44e-04	6.31e-04	9.65e-04	345.6	3.0	1.0
EP (fixed α)	4.63e-04	1.43e-04	9.29e-04	3.6	685.3	1.1
EP2	8.46e-04	1.83e-04	9.87e-04	1.3	18.2	1.0
EP2 (constr.)	5.51e-04	6.59e-04	9.69e-04	2.4	1.9	1.0
EP2 (fixed α)	6.25e-04	6.29e-04	9.27e-04	3.2	1.7	1.1
HUP	7.55e-04	5.02e-04	9.59e-04	1.5	3.8	1.1
HUP (constr.)	7.23e-04	2.64e-04	8.49e-04	5.9	312.9	1.2
K	1.43e-04	1.48e-04	9.99e-04	857.3	208.2	1.0
Prop.	5.58e-04	5.43e-04	7.89e-04	7.3	4.0	1.4
Oh	1.44e-04	7.65e-04	8.80e-04	319.3	4.4	1.2

Table 5.3: Average weight variances and effective sample sizes for the results from the PMC sampler on the five target data set.

By comparing the MAP probability to its MCMC-derived counterpart, it is apparent that the sample lies in a low probability region. Thus, the ESS failed to diagnose the mismatch between proposal and target distribution. In contrast to what we saw using the one target data set, there is a weak negative correlation ($r = -0.089$) between the expected score and the average ESS in these experiments.

Since the quality of the samples produced by the PMC algorithm was not satisfactory, we also looked at whether a different allocation of resources would improve them. First, we considered a sample size of 7000 computed using a single PMC iteration. This effectively evaluated the effectiveness of the initialization scheme itself as the proposal distribution and took the iterative updates out of the equation. Looking at our canonical example, the scan-constrained DP prior, the resulting estimate of K was 6.00 ± 0.01 with an estimated combined score of 0.71 ± 0.00, which represents a substantial improvement. The variance of the weights, however, took the maximum value and the ESS was just 1.0, which indicates total degeneracy. Next, we considered a smaller sample size with the remaining resources allocated to the iterations. A sample size of 100 with 70 iterations was used. Continuing with our example, K was 72 ± 0.00 with an estimated combined score of just 0.065 ± 0.00. As our preliminary investigations indicated, small samples appear to be detrimental to the performance. Finally, we considered keeping the original sample size of 1000 but increasing the number of iterations

to 70. The resulting estimate of K was 7.00 ± 0.00 with an estimated combined score of 0.61 ± 0.00. The ESS in this case was 3.5. This lends support to the notion in the PMC literature that often a relatively small number of iterations is sufficient and suggests that any additional resources should also go to support an increase in the sample size. It also showed that an increase in the number of PMC iterations is insufficient to stave off weight collapse.

Switching to the alternative implementation of the proposal distribution again tended to improve the combined scores and estimates compared to the default. However, the weakly resolved target 5 was missed somewhat more frequently. Compared to their MCMC-derived counterparts, the combined scores from the PMC sampler using the alternative implementation of the proposal distribution were actually slightly lower in just over half of the experiments. With respect to the cluster size estimates, the performance of the DP, EP, and EP2 nonparametric priors using a small fixed α and the scan-constrained HUP prior returned to parity with their MCMC-derived counterparts. Here too there was a tendency to slightly underestimate K. Rather unexpectedly, the "K" prior, which arguably expresses some questionable beliefs about the distribution over cluster labels, was among the configurations to produce the correct estimate of the number of clusters. Although the variance estimates for the number of clusters remained near zero in all but two of the experiments, the ESS's generally improved. They remained, however, unacceptably small. The effective sample sizes for the DP and EP priors with a small fixed α, and the scan-constrained HUP prior have improved considerably, as was the case in the experiments using the one target data set. Note that despite having the fifth highest ESS, the sample produced using the "K" prior again lay in a low probability region as it did when using the default proposal distribution. Since the trend toward higher ESS values after switching to the alternative implementation of the proposal distribution was also observed in the experiments using the one target data set, this suggests that modifying the proposal distribution is indeed a potential remedy.

Finally, let us look at the performance of the PMC sampler using the k-add mixture proposal distribution. Table D.19 shows that the expected combined scores only improved in four of the experiments. Overall, there was a tendency to slightly underestimate the number of targets, similar to what we saw when using the alternative implementation of the proposal distribution, but here it was true for all of the cluster

label priors except for the "K" prior. Even that one only overestimated the number of clusters by just one. This insensitivity to the choice of prior fits with our earlier notion of a strong likelihood component of the model. The results here also suggest that those obtained when using the alternative implementation of the proposal distribution are not entirely unreasonable because even with the extra freedom of movement afforded by k-add mixture proposal distribution, the sampler avoided configurations with excessive numbers of targets. Unfortunately, as we can see from table 5.3 , the problem of weight collapse was severe, with almost total degeneracy in all of the experiments. However, the results from the one target data set suggest that the problem is not endemic to the k-add mixture proposal distribution. For the critical metric of the number of targets, all variance estimates were negligible. Thus, further development is needed before this proposal distribution would be suitable for surveillance applications.

5.5.6 Field Test

The models and samplers were also evaluated using data collected from a field test of a commercial surveillance radar. In this case, we do not have ground truth and thus can only make a qualitative assessment of the results. Known targets were carrying handheld GPS receivers which recorded their tracks with moderate precision. This gives us a good idea of what to expect, at least in the vicinity of the targets, but it is impossible to reliably determine the origin of each measurement based on this information. Despite these limitations, the value of such experiments is to see what happens when the algorithms are exposed to the realistic complications of non-uniform false alarms, unknown spatially varying P_D, and imperfect measurement models.

In this data set, three walkers are moving along a trail in rolling terrain at a range of approximately 3.5 km. They are traveling together, as a group, which introduces an additional complication. The resolution of radar is insufficient to resolve the targets individually, so the best we can expect from the tracker is that it identifies a single target corresponding to the group. There may well be occasional instances where more than one of the targets is detected in a given scan, but that is unlikely. What is more likely is that the presence of multiple targets within a resolution cell will effectively be a source of additional measurement error. The data set consists of approximately 30 minutes worth of measurements collected at 20 RPM ($T = 3\,\mathrm{s}$). The area of the surveillance region is $2.3\,\mathrm{km}^2$.

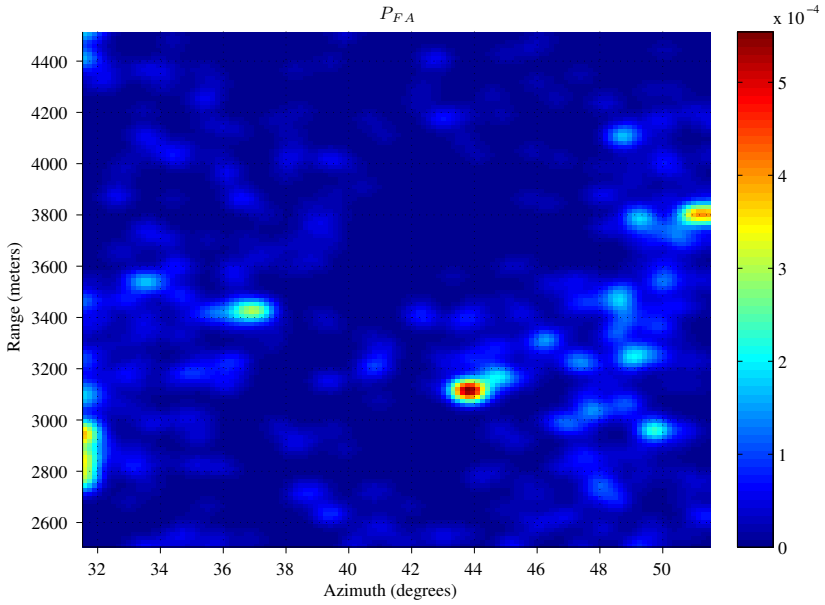


Figure 5.28: Probability of false alarm for the field test data.

With an average of just 2.33 detections per scan, the false alarm rate is significantly lower than what we have considered in the previous experiments. This is because a low false alarm rate is expected for a surveillance radar as traditional trackers tend to break down in the presence of high number of false alarms. Of course, this usually comes at the cost of many missed detections. Recall that one of the objectives of this research was to develop a tracker that can effectively tolerate more false alarms in order to maximize the probability of detection. Figure 5.28 shows the estimated of probability of false alarms over the surveillance region, which is clearly non-uniform. It appears that something in the environment, perhaps wind-blown foliage, is confusing the detector's threshold selection logic which leads to these "hot spots". Such behavior is not uncommon yet we have made no attempt to mitigate the nonuniformity by preprocessing the detections, preferring instead to work with the raw data. The associative tracking model was still configured based on the assumption of uniformly distributed false alarms in order to deliberately maintain the mismatch between the false alarm model and the data.

A P_D model estimated based on the line of sight visibility, given the terrain and the signal propagation, was also used. It is not known how accurate the model is because it has not been validated empirically. Figure 5.29 shows a portion of the model

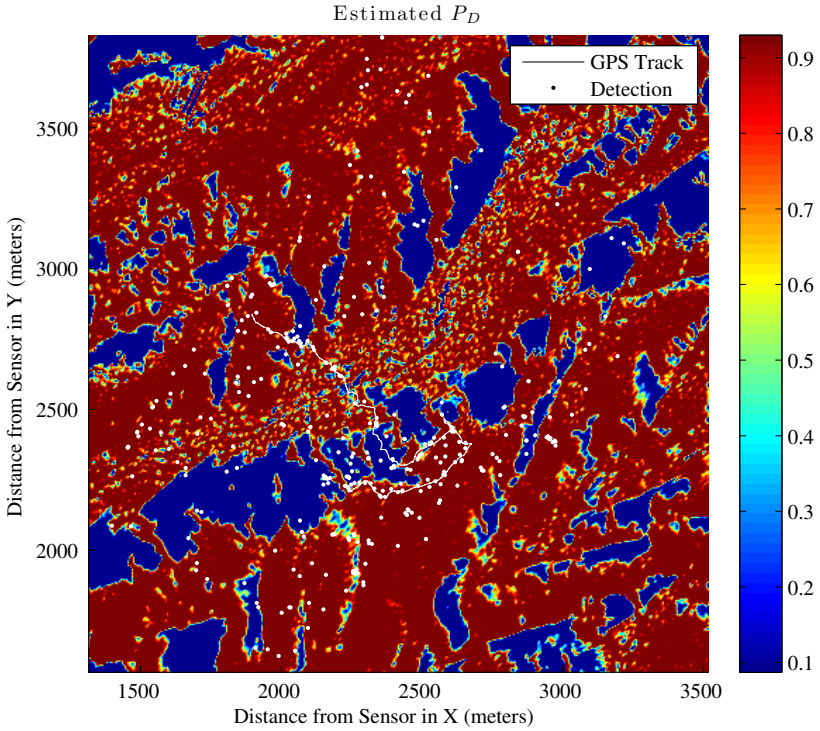


Figure 5.29: Estimated P_D plus the detections and GPS track for a subset of the field test data.

for the region containing the walkers. The GPS track and detections are also included for reference. Note how the targets skirt the edge of the (presumably) detectable regions in several places and then pass through a large region of negligible probability of detection. The large gap represents a dropout of just under three minutes. Since we do not expect the probability of detection to be unity anywhere and because some of regions of negligible probability of detection may in fact be mischaracterized, the model used by the tracker was rescaled so that $P_D \in [0.07, 0.93]$.

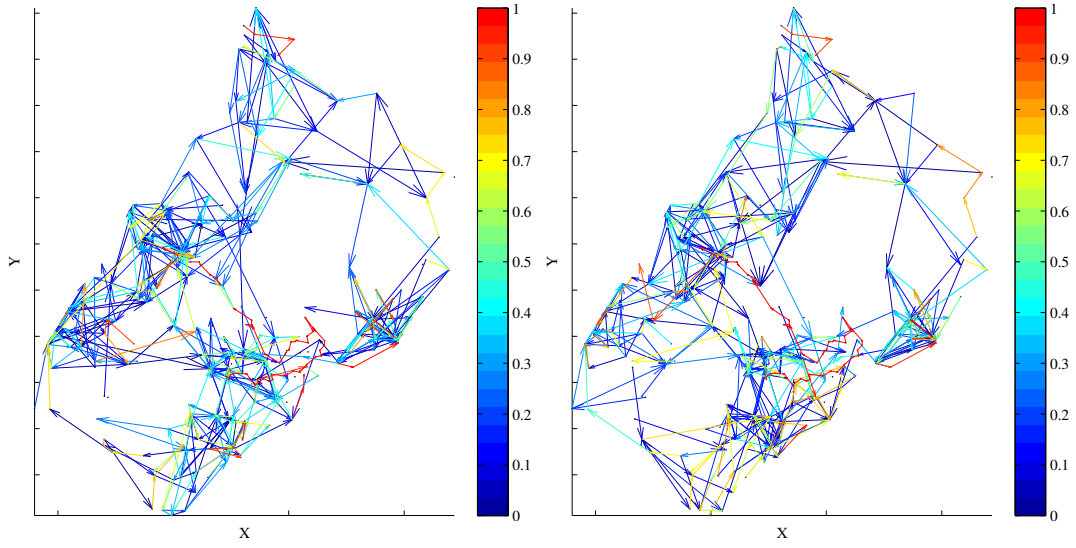
To handle the sparse detections, a path constrained dynamic model with Wiener process speed (PSH-Path/C1) was used to construct the feasible graph. The maximum out-degree of each vertex was again limited to 5. Lacking ground truth, we made a qualitative assessment of the coverage and deemed it to be sufficient.

Tables D.21–D.25 contain the summary statistics from the experiments. Note that they do not include the scores or score-based metrics that were so critical in our earlier evaluations because we no longer have the necessary information to compute them. Thus, here we have relied primarily on qualitative assessments of the weighted-

edge-sample plots and MAP estimates of Z in their absence.

Just as we saw in the previous experiments based on the simulated measurements, the MCMC sampler using the default implementation of the proposal distribution had a tendency to grossly overestimate the number of targets. The false alarm rate was consistently underestimated as we would expect under the circumstances. Despite far fewer total measurements than were in the one target data set, the expected number of targets was larger in all but one of these experiments. Although the DP prior using a small fixed α once again produced the best estimates, the proposition that there are 5.8 ± 0.74 targets is not very reasonable. Figure 5.30 shows the results from a selection of priors that consists of the DP-based, scan-constrained HUP, and "Oh" priors. The corresponding MAP estimates of Z are shown in figure 5.31. We can see that the weighted-edge-sample plots paint a somewhat confusing picture of what is happening in the surveillance region. However, in these examples the tracker appears to have reliably identified the target measurements in the regions of high P_D. The MAP estimates of Z reveal that the DP-based priors produced a track that matches up well with the GPS track, unfortunately it is buried among numerous tracks consisting of noise. The the scan-constrained HUP and "Oh" priors, on the other hand, failed to contain a good match for the true track in the MAP estimate of Z. They both produced assignments in which the track was broken into fragments. Despite failing to keep the track intact, the "Oh" prior did a remarkable job of suppressing spurious associations elsewhere in the region. Even the activity to the right of the target, which most likely consists entirely of noise, produced only low-probability associations unlike what we saw with the other priors. In the experiments not pictured, the nonparametric priors needed the small fixed α before their preferences were strong enough to keep the track intact; otherwise it was broken into fragments. The modest reduction in α achieved by introducing the scan constraints to the nonparametric priors led to very little improvement. Of the remaining priors, only the HUP and "Prop." maintained the track, but buried under substantial noise.

Using the alternative implementation of the proposal distribution with the MCMC sampler improved the output considerably. As we can see from table D.22, in almost half of the experiments the expected number of targets is, for lack of a better word, what we expected: 1.0. Only the scan-constrained HUP, "K", and "Prop." priors yielded significant overestimates with numerous high-probability erroneous associations.



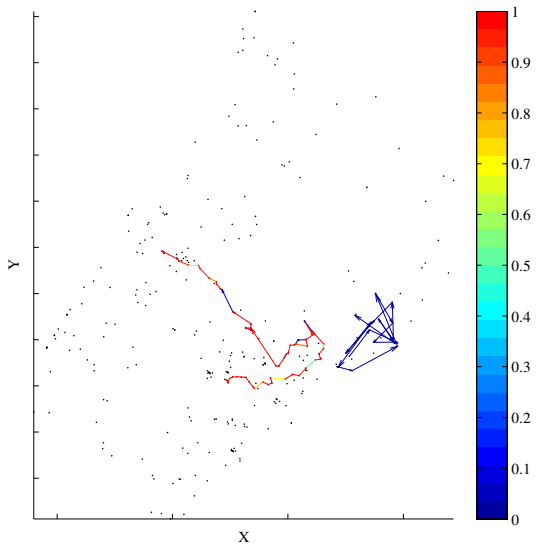
(a) DP prior.

(b) Scan-constrained DP prior.

(c) DP prior with $\alpha = 1 \times 10^{-6}$.

(d) Scan-constrained HUP prior.

Figure 5.30: Output from selected priors. The results were derived from an MCMC sample of the field test data set.

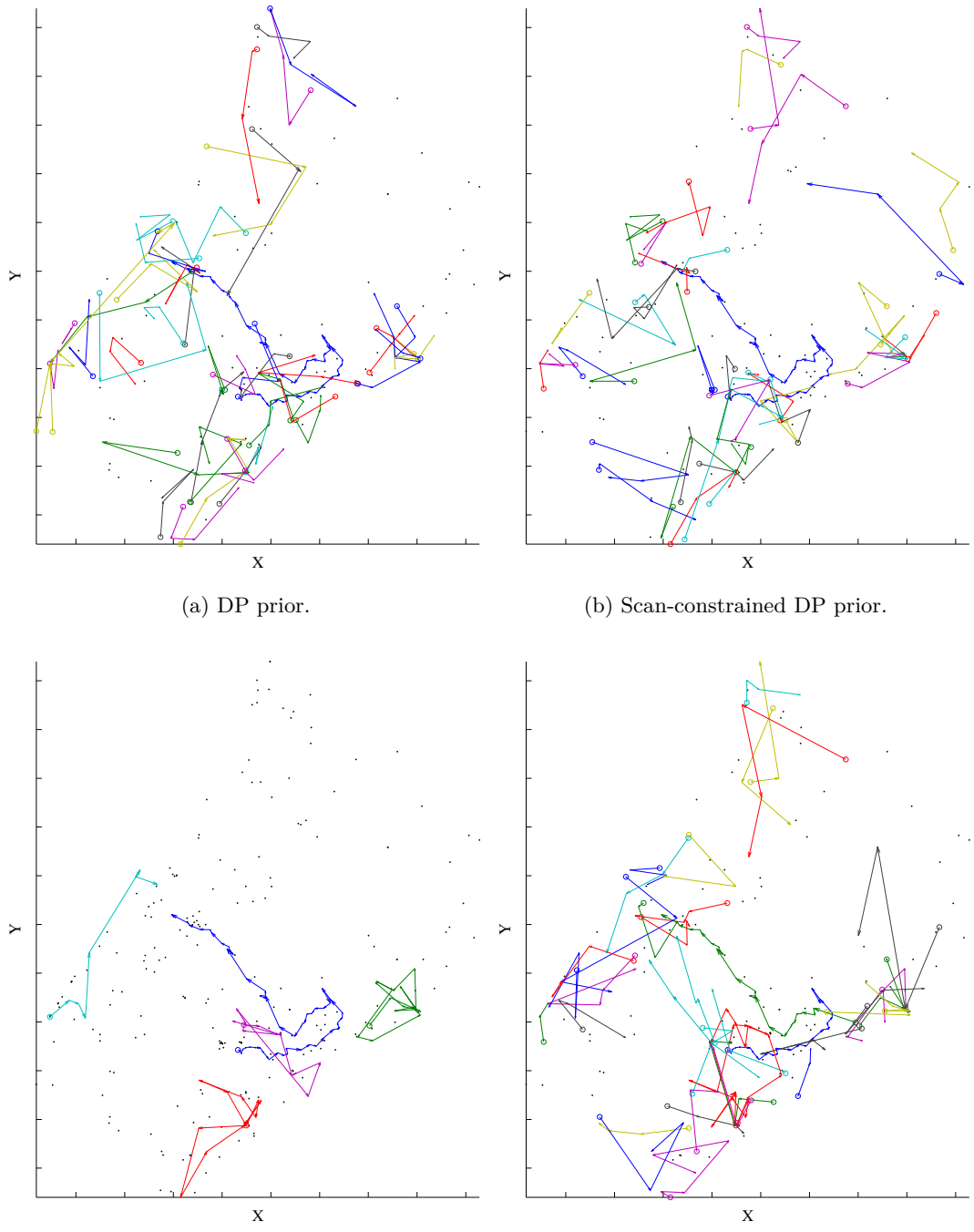


(e) "Oh" prior.

Continuation of Figure 5.30: Output from selected priors. The results were derived from an MCMC sample of the field test data set.

The variance estimates appear to be understated in all experiments. Compared to the default proposal distribution, the switch to the alternative implementation entailed a 5-10% drop in the acceptance rate to an average of approximately 0.26. Figures 5.32 and 5.33 show the weighted-edge-sample plots and the MAP estimates of Z, respectively, for the same selection of priors that we looked at earlier. A visual inspection of the weighted-edge-sample plots confirms that the high probability edges indeed agree with the GPS data and the results are encouraging. Of all the priors, only the scan-constrained HUP and "K" failed to keep the track intact. Once again, the MCMC sampler using the alternative implementation of the proposal distribution failed to explore any of the higher probability configurations that were reached using the default proposal distribution. As was the case for the earlier experiments based on the simulated measurements, this was to our advantage but still a cause for concern.

When the samples were generated using the PMC algorithm, we can see from tables D.23 and D.24 that the results followed the same basic pattern that was observed in the other data sets. That is: generally lower expected target counts compared to the corresponding MCMC experiments for the default proposal distribution, a further reduction toward the (nominally) true number of targets when each move considered maintaining the status quo, a further reduction of the expected number of targets when



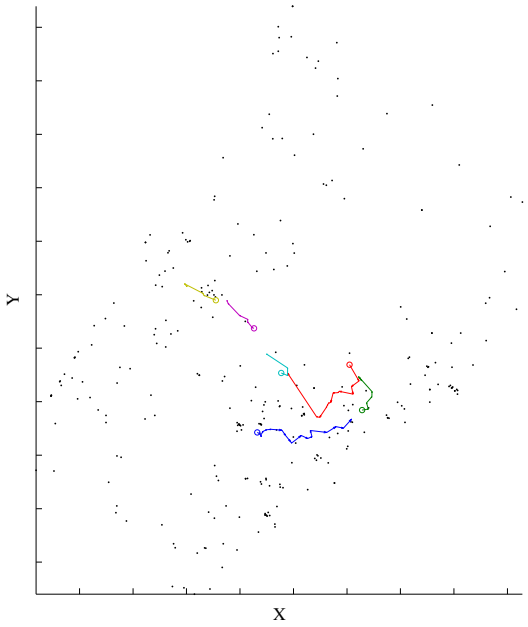
(a) DP prior.

(b) Scan-constrained DP prior.

(c) DP prior with $\alpha = 1 \times 10^{-6}$.

(d) Scan-constrained HUP prior.

Figure 5.31: MAP estimates of Z from selected priors. The results were derived from an MCMC sample of the field test data set.

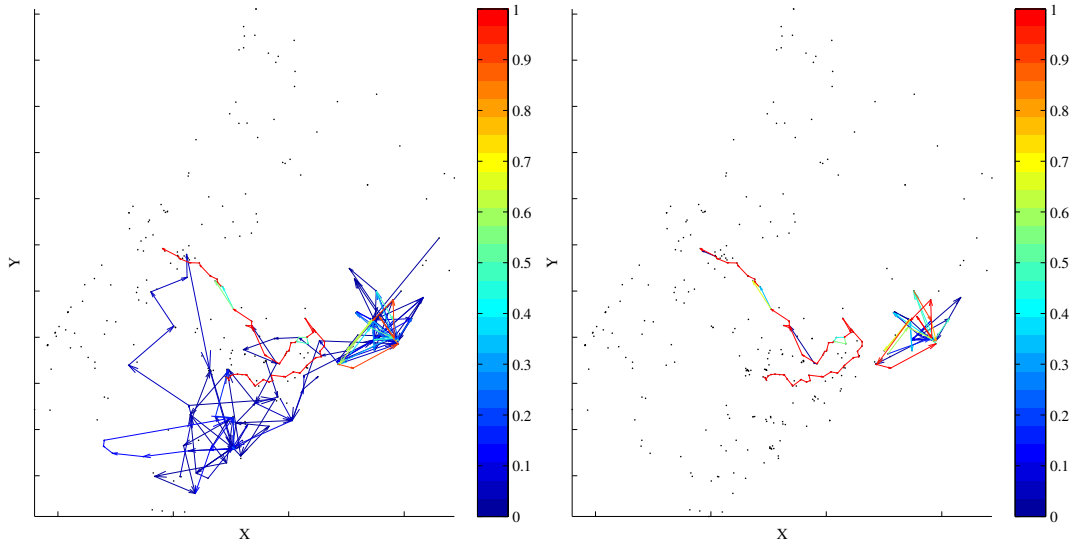


(e) “Oh” prior.

Continuation of Figure 5.31: MAP estimates of Z from selected priors. The results were derived from an MCMC sample of the field test data set.

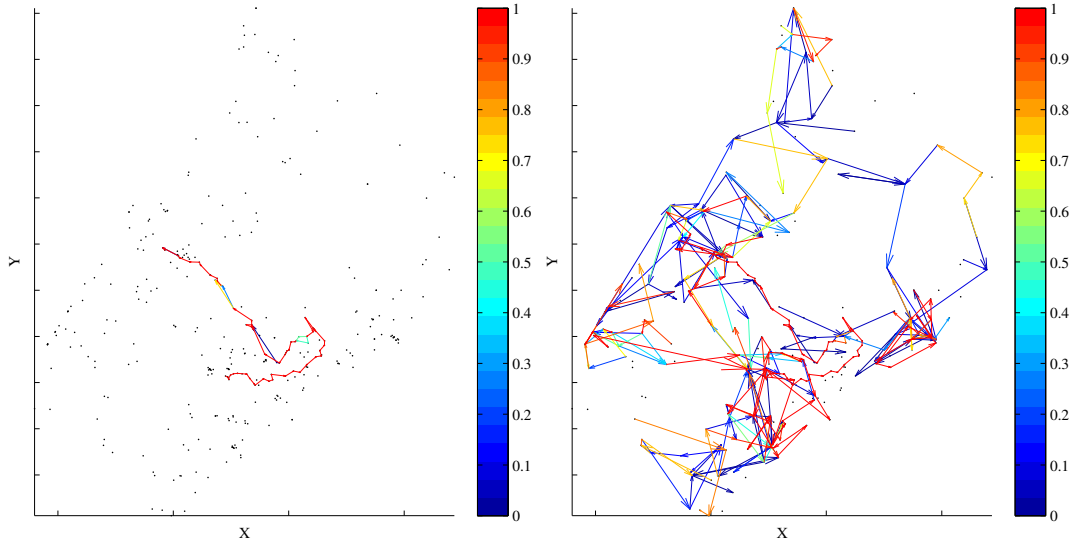
the k-add mixture proposal distribution was used, and the collapse of the variance estimates in all three cases. Note that a few experiments did produce non-negligible estimates of the variance in the number of targets, although none were plausible.

Table 5.4 shows the average weight variances and average effective sample sizes (ESS) for all of the PMC experiments. For the default implementation of the proposal distribution, the results were generally poor and similar to what we saw using the one target data set consisting of simulated measurements. The HUP and “K” priors stand out as exceptions in that regard, but their relatively high ESS’s appear to be misleading in the same sense that we discussed in the previous section. Switching to the alternative implementation of the proposal distribution did not lead to the same kind of improvements that we have seen before. Only one prior, the EP with a small fixed α, improved considerably. However, the ESS values grew in just over half of experiments so in this sense there is still a weak trend toward improvement of the samples. For the k-add mixture proposal distribution, three priors managed to achieve more than 20% of the optimal ESS, but all remained unacceptably small with most indicating almost total degeneracy.



(a) DP prior.

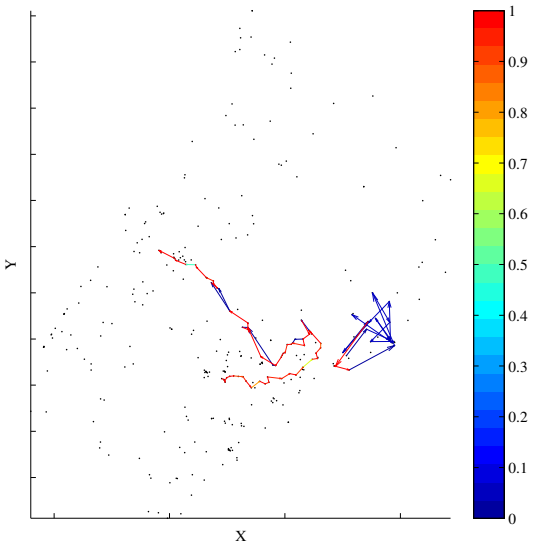
(b) Scan-constrained DP prior.



(c) DP prior with $\alpha = 1 \times 10^{-6}$.

(d) Scan-constrained HUP prior.

Figure 5.32: Output from selected priors when the individual moves in the proposal distribution were allowed to propose remaining in the current state. The results were derived from an MCMC sample of the field test data set.

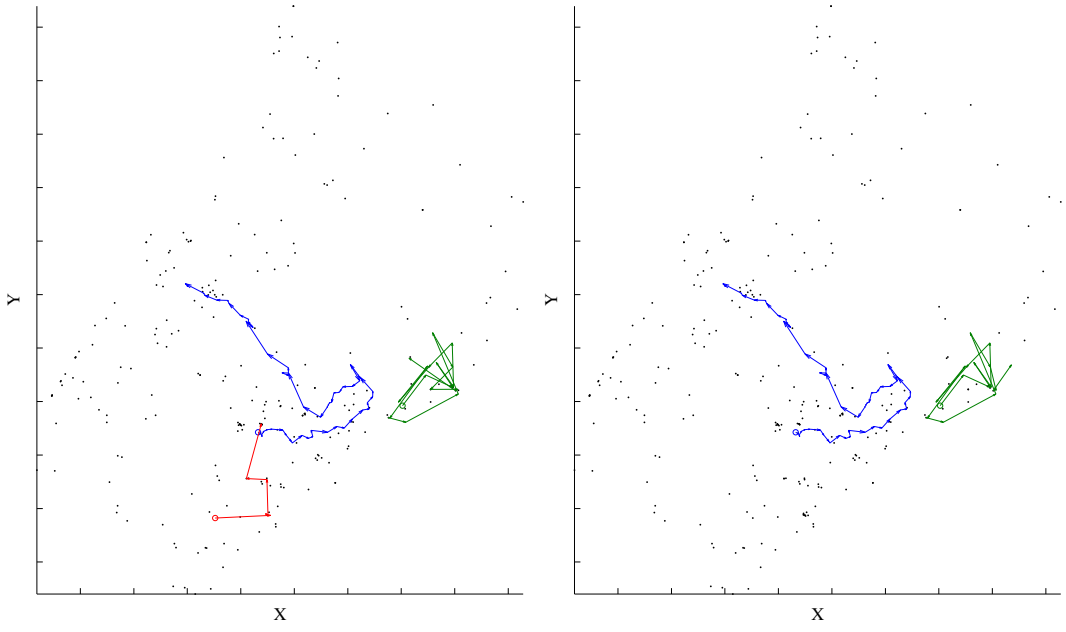


(e) “Oh” prior.

Continuation of Figure 5.32: Output from selected priors when the individual moves in the proposal distirbution were allowed to propose remaining in the current state. The results were derived from an MCMC sample of the field test data set.

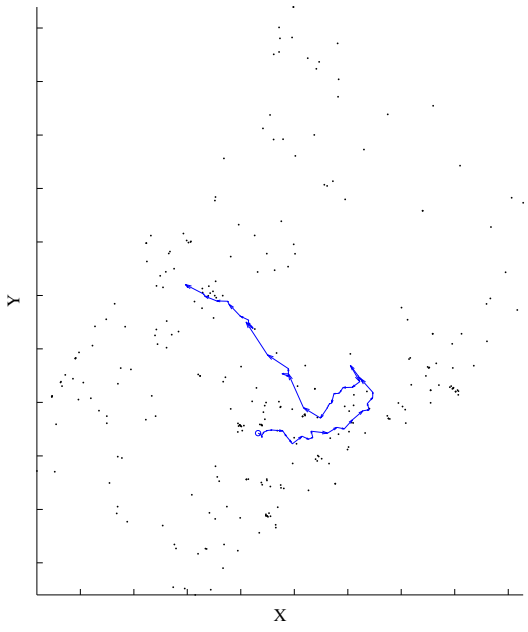
Prior	**Avg. Variance**			**Avg. ESS**		
	Default	Status Quo	K-add	Default	Status Quo	K-add
DP	2.87e-04	5.65e-04	9.70e-04	6.9	2.7	1.0
DP (constr.)	4.31e-04	3.76e-04	6.18e-04	5.9	3.9	2.0
DP (fixed α)	4.05e-04	1.97e-04	1.38e-05	4.0	7.0	258.5
UP	5.24e-04	5.91e-04	8.83e-04	3.1	18.6	1.2
UP (constr.)	6.48e-04	5.34e-04	9.20e-04	4.1	4.9	1.1
UP (fixed α)	7.04e-04	6.62e-04	4.53e-04	2.7	1.8	2.9
EP	7.41e-04	6.81e-04	1.73e-04	1.8	2.8	23.7
EP (constr.)	6.01e-04	7.48e-04	6.03e-05	12.8	1.7	20.5
EP (fixed α)	4.40e-04	1.68e-05	5.73e-05	3.1	349.6	245.7
EP2	5.35e-04	5.25e-04	8.93e-04	2.8	2.3	1.1
EP2 (constr.)	5.87e-04	5.14e-04	3.89e-04	2.0	2.7	46.7
EP2 (fixed α)	5.92e-04	5.95e-04	3.31e-05	2.4	29.9	221.9
HUP	1.43e-04	6.43e-04	5.38e-04	857.3	2.3	2.8
HUP (constr.)	4.55e-04	5.03e-04	8.27e-04	3.1	10.6	1.4
K	1.07e-04	4.20e-04	8.89e-04	818.9	5.2	1.2
Prop.	4.56e-04	5.34e-04	9.40e-04	4.5	2.2	1.1
Oh	1.51e-04	1.76e-04	4.54e-05	45.2	71.4	80.0

Table 5.4: Average weight variances and effective sample sizes for the results from the PMC sampler on the field test data set.

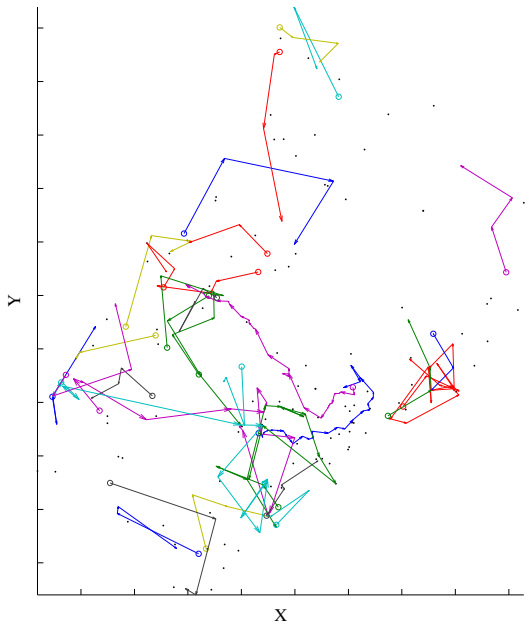


(a) DP prior.

(b) Scan-constrained DP prior.

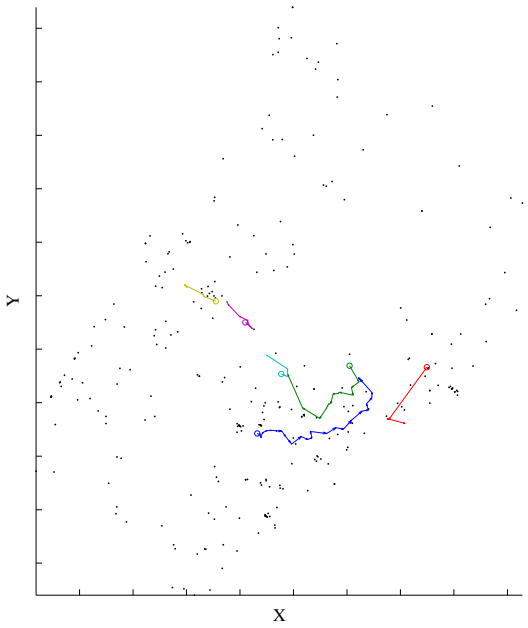


(c) DP prior with $\alpha = 1 \times 10^{-6}$.



(d) Scan-constrained HUP prior.

Figure 5.33: MAP estimates of Z from selected priors when the individual moves in the proposal distribution were allowed to propose remaining in the current state. The results were derived from an MCMC sample of the field test data set.

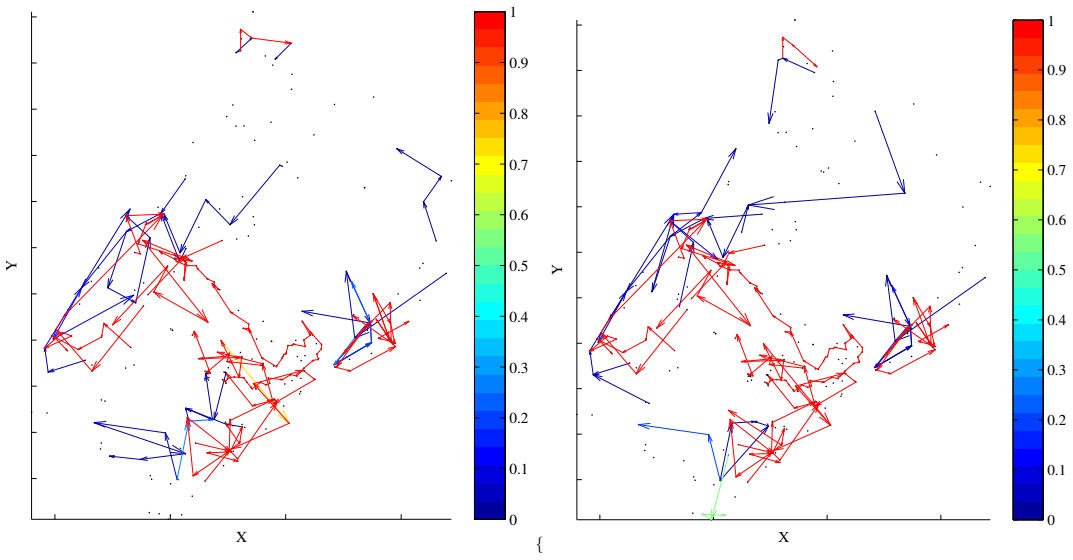


(e) “Oh” prior.

Continuation of Figure 5.33: MAP estimates of Z from selected priors when the individual moves in the proposal distirbution were allowed to propose remaining in the current state. The results were derived from an MCMC sample of the field test data set.

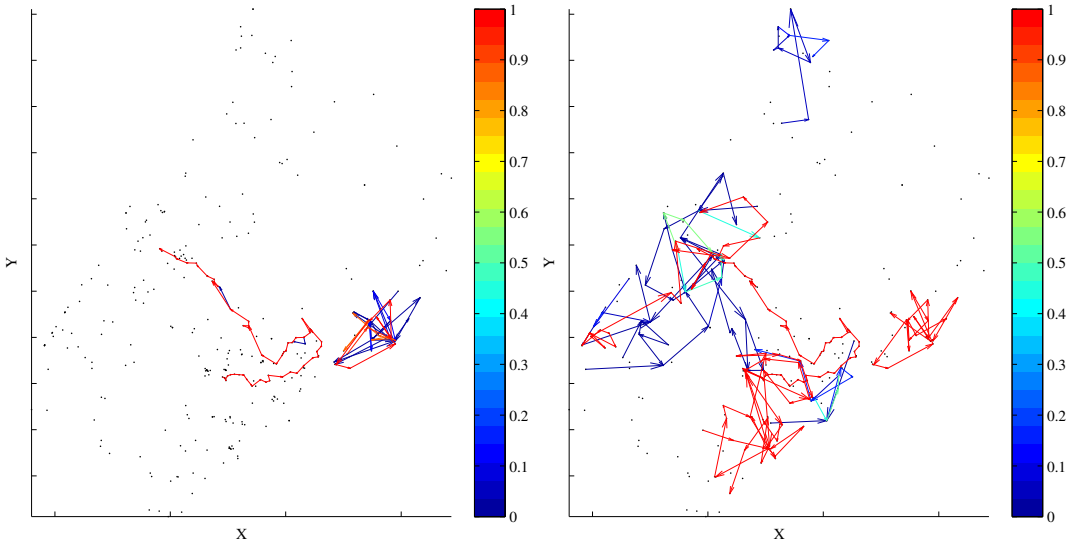
For comparison with the earlier results from the field test data set, figures 5.34 and 5.35 show the weighted-edge-sample plots and the MAP estimates of Z, respectively, for the selected priors when the default proposal distribution was used; figures 5.36 and 5.37 contain the corresponding plots for the case of the alternative implementation of the proposal distribution. For the sake of completeness, figures 5.38 and 5.39 contain the corresponding plots for the case of the k-add mixture proposal distribution.

Note how compared to figure 5.30, the weighted-edge-sample plots in figure 5.34 show supreme confidence in many of the erroneous associations, which presents a very misleading picture of what is happening within the surveillance region. In almost all of the PMC experiments, the MAP estimates of Z managed to contain a good match with the GPS track along with varying numbers of erroneous tracks. The "Oh" prior produced MAP estimates that fragmented the track when using the default and alternative implementations of the proposal distribution. The EP prior, both unconstrained and using a small fixed α, also failed to keep the track intact when using the alternative implementation of the proposal distribution. Only the DP prior using a small fixed α produced reasonable results when using either the default or alternative implementation of the proposal distribution. Overall, the estimates and posterior probabilities were quite similar across all experiments using those two proposal distributions. With the k-add mixture proposal distribution, the estimates improved and the posterior probabilities dropped moderately. All MAP estimates of Z contained a good match with the GPS track in this case with only the HUP and scan-constrained HUP priors producing significant numbers of spurious tracks. Seven of the experiments yielded results that appeared to be perfect, while five others identified the true track plus a well-separated (presumably) erroneous track. Despite these encouraging results, we should not forget that the uncertainty was very poorly characterized. Figure 5.38d shows a rather extreme example of the consequences with all associations regarded as equally likely.



(a) DP prior.

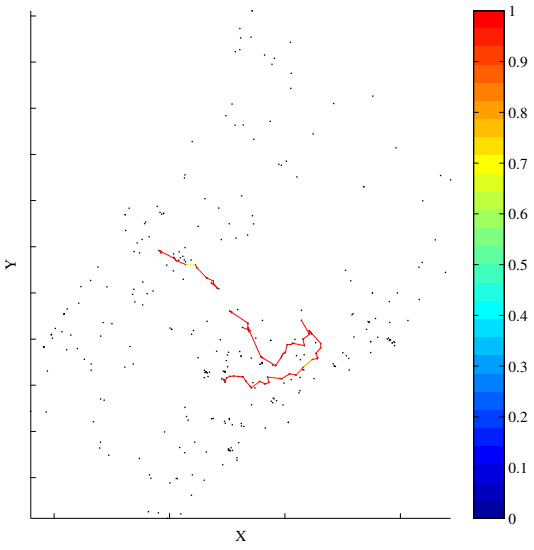
(b) Scan-constrained DP prior.



(c) DP prior with $\alpha = 1 \times 10^{-6}$.

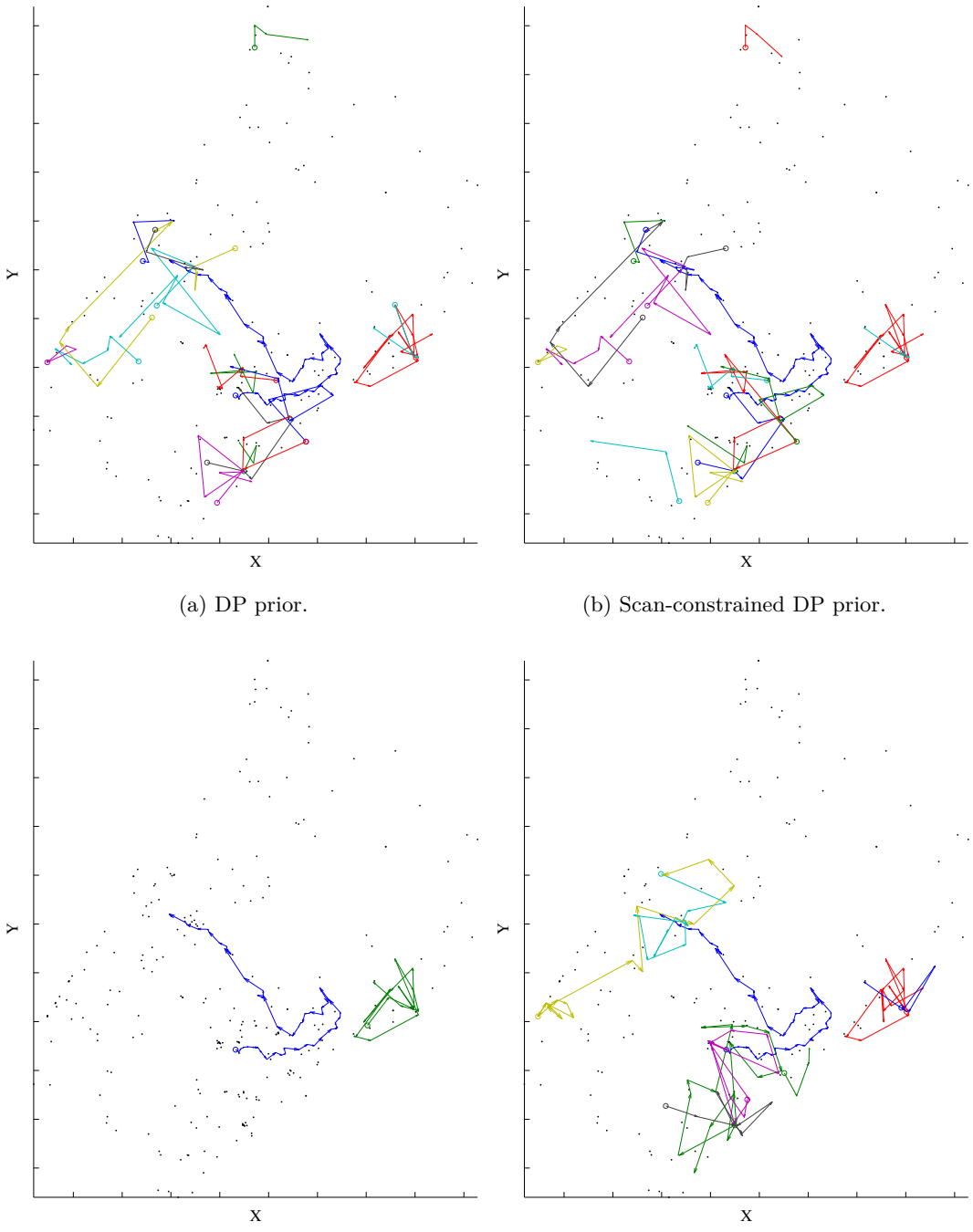
(d) Scan-constrained HUP prior.

Figure 5.34: Output from selected priors. The results were derived from an PMC sample of the field test data set.



(e) “Oh” prior.

Continuation of Figure 5.34: Output from selected priors. The results were derived from an PMC sample of the field test data set.



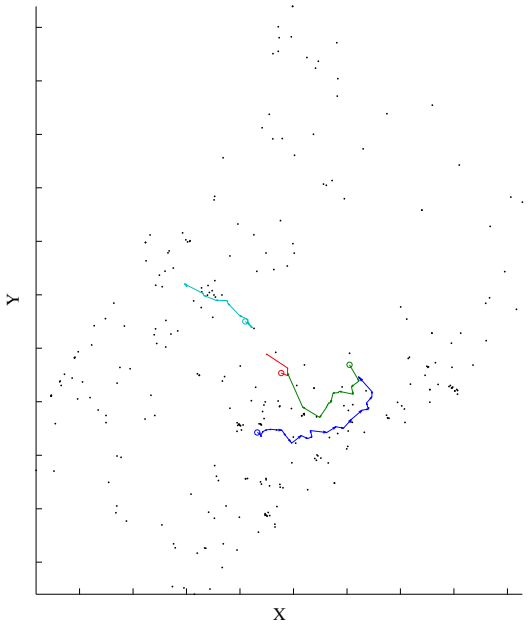
(a) DP prior.

(b) Scan-constrained DP prior.

(c) DP prior with $\alpha = 1 \times 10^{-6}$.

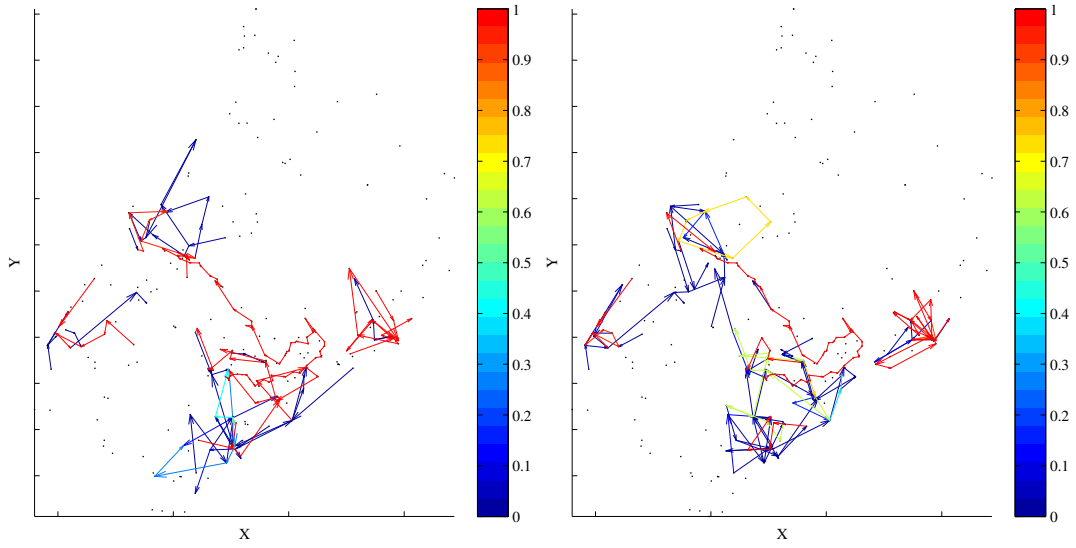
(d) Scan-constrained HUP prior.

Figure 5.35: MAP estimates of Z from selected priors. The results were derived from an PMC sample of the field test data set.



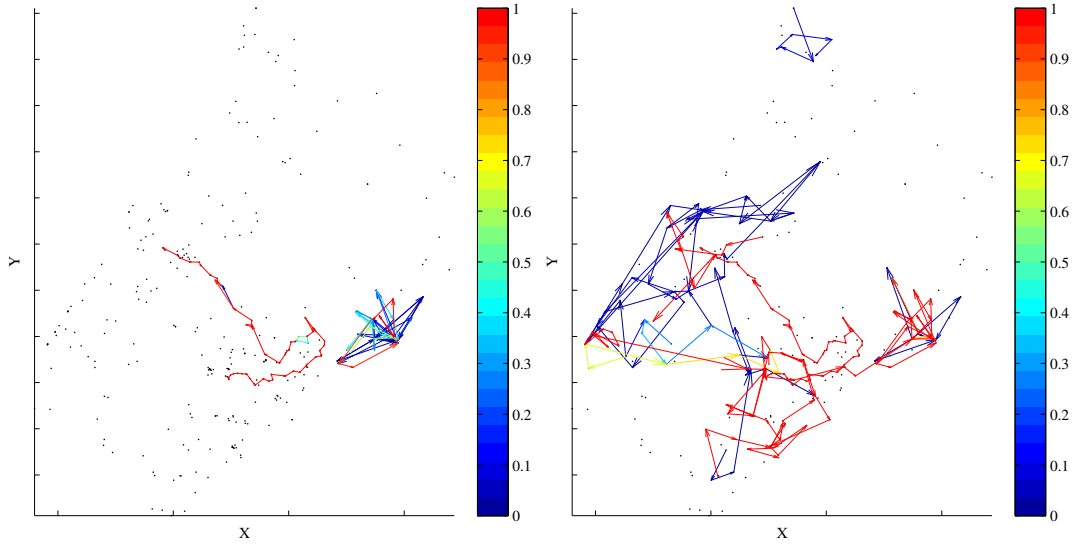
(e) “Oh” prior.

Continuation of Figure 5.35: MAP estimates from selected priors. The results were derived from an PMC sample of the field test data set.



(a) DP prior.

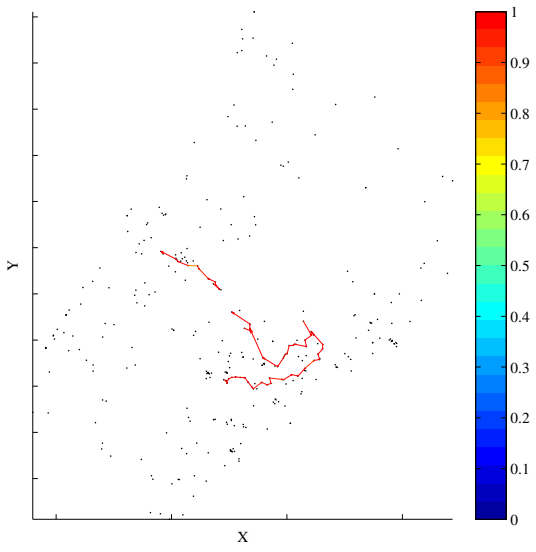
(b) Scan-constrained DP prior.



(c) DP prior with $\alpha = 1 \times 10^{-6}$.

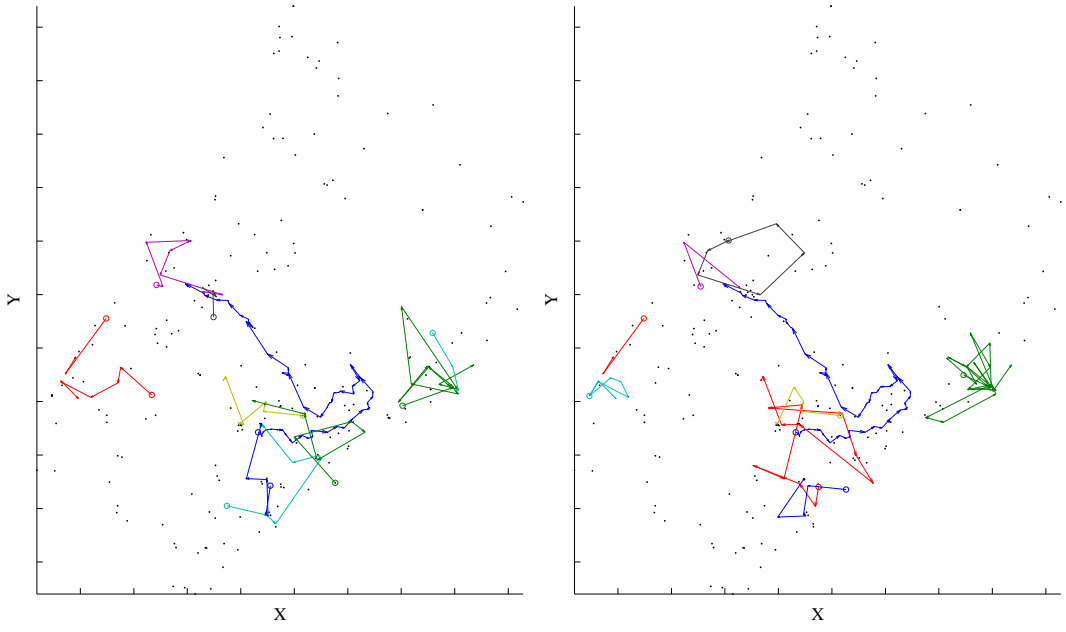
(d) Scan-constrained HUP prior.

Figure 5.36: Output from selected priors when the individual moves in the proposal distribution were allowed to propose remaining in the current state. The results were derived from an PMC sample of the field test data set.



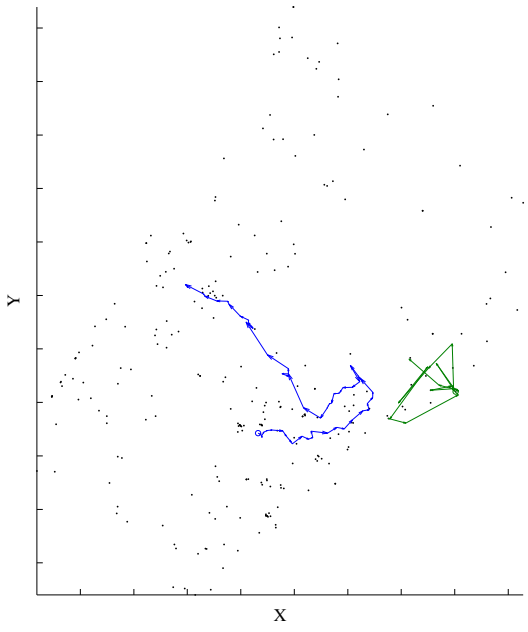
(e) “Oh” prior.

Continuation of Figure 5.36: Output from selected priors when the individual moves in the proposal distirbution were allowed to propose remaining in the current state. The results were derived from an PMC sample of the field test data set.

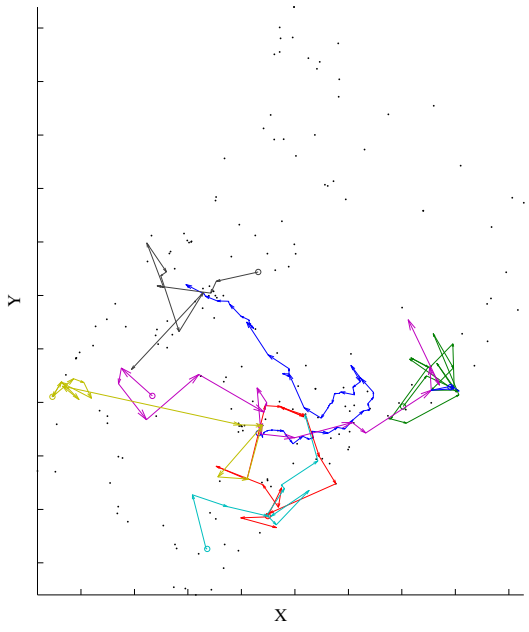


(a) DP prior.

(b) Scan-constrained DP prior.

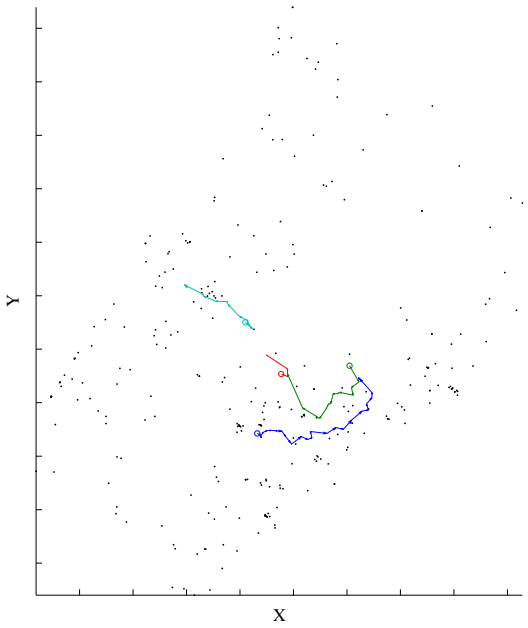


(c) DP prior with $\alpha = 1 \times 10^{-6}$.



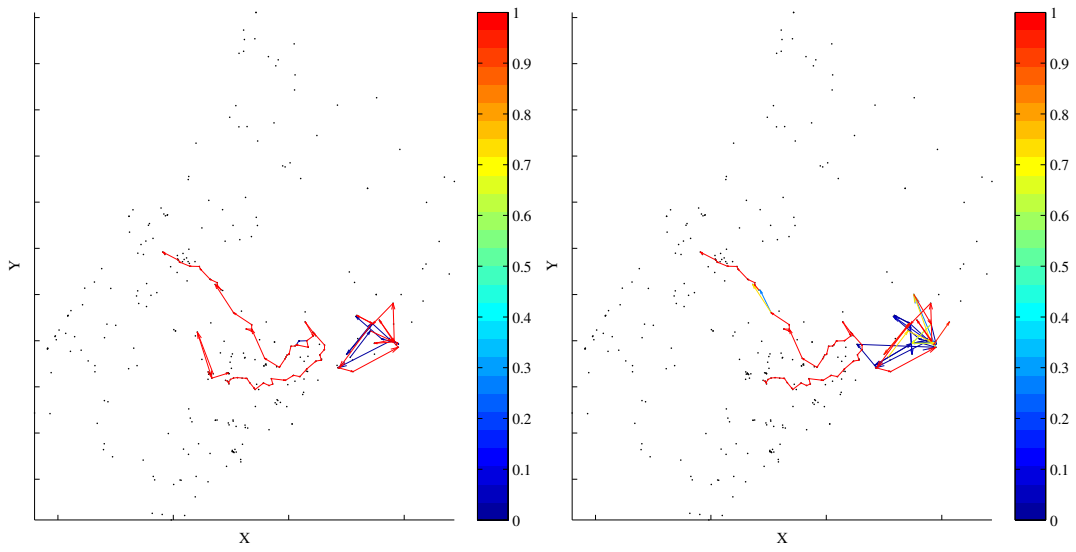
(d) Scan-constrained HUP prior.

Figure 5.37: MAP estimates of Z from selected priors when the individual moves in the proposal distribution were allowed to propose remaining in the current state. The results were derived from an PMC sample of the field test data set.



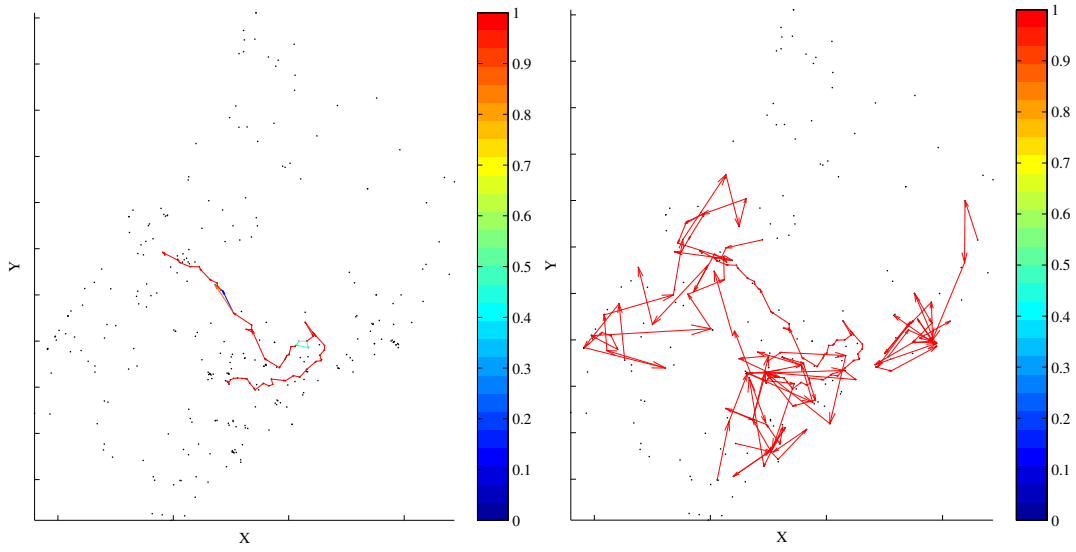
(e) “Oh” prior.

Continuation of Figure 5.37: MAP estimates of Z from selected priors when the individual moves in the proposal distirbution were allowed to propose remaining in the current state. The results were derived from an PMC sample of the field test data set.



(a) DP prior.

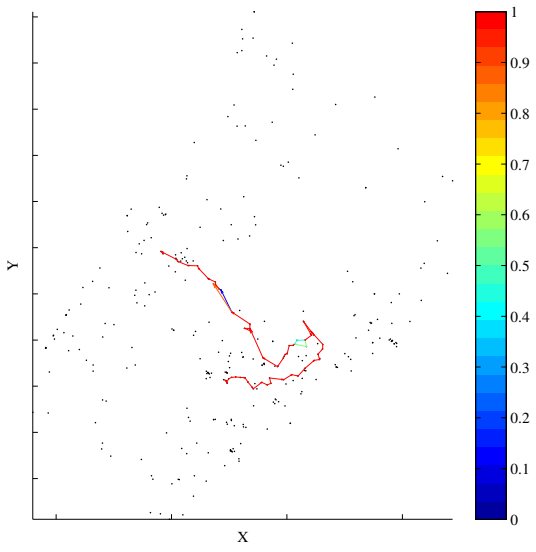
(b) Scan-constrained DP prior.



(c) DP prior with $\alpha = 1 \times 10^{-6}$.

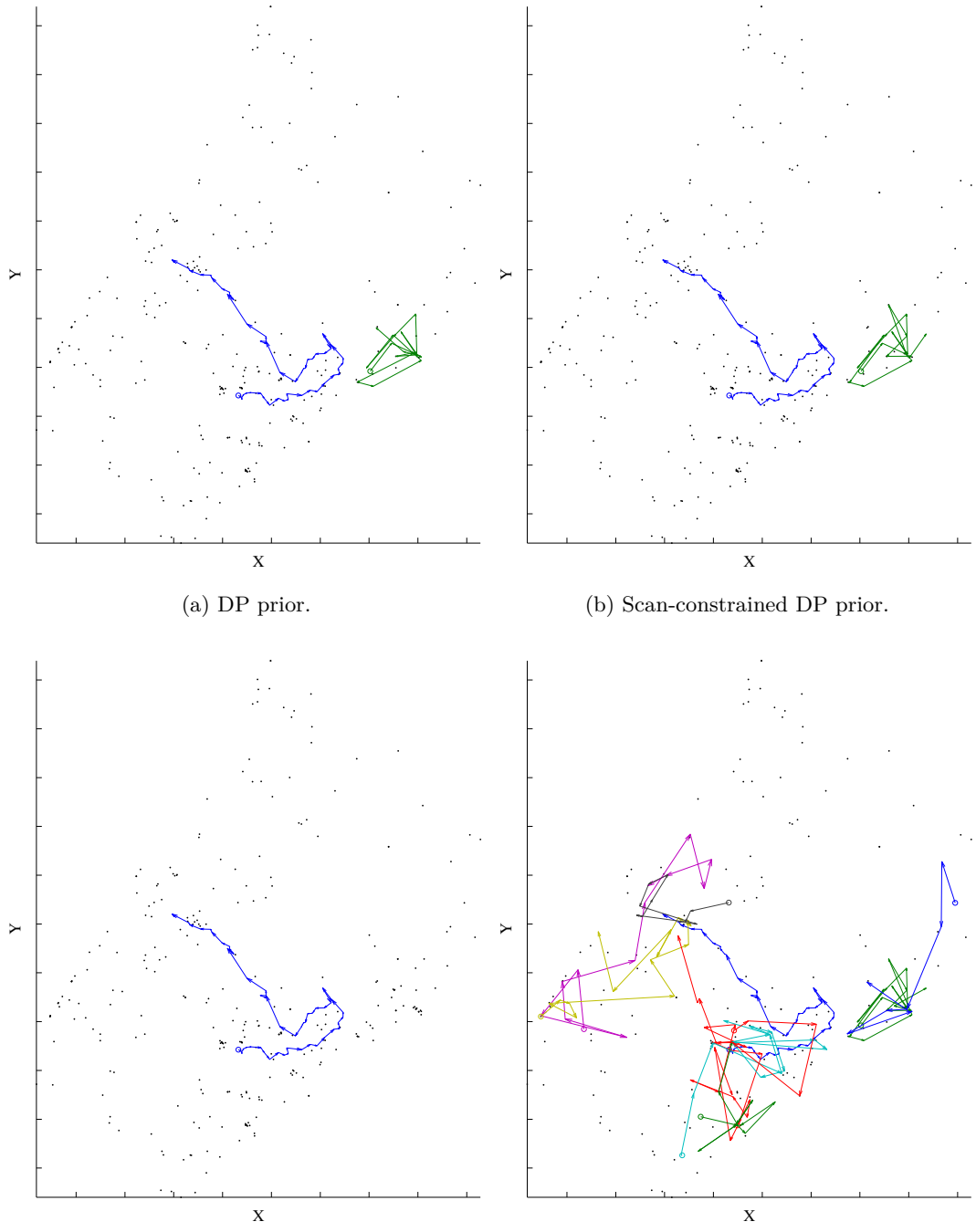
(d) Scan-constrained HUP prior.

Figure 5.38: Output from selected priors using the k-add mixture proposal distribution. The results were derived from an PMC sample of the field test data set.



(e) “Oh” prior.

Continuation of Figure 5.38: Output from selected priors using the k-add mixture proposal distribution. The results were derived from an PMC sample of the field test data set.



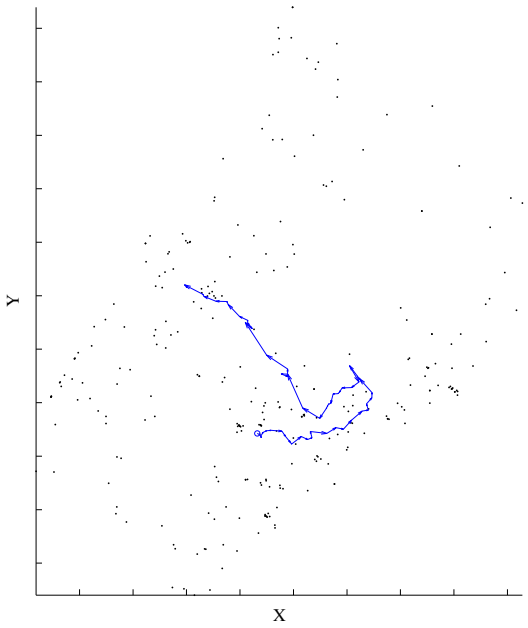
(a) DP prior.

(b) Scan-constrained DP prior.

(c) DP prior with $\alpha = 1 \times 10^{-6}$.

(d) Scan-constrained HUP prior.

Figure 5.39: MAP estimates of Z from selected priors using the k-add mixture proposal distribution. The results were derived from an PMC sample of the field test data set.



(e) “Oh” prior.

Continuation of Figure 5.39: MAP estimates of Z from selected priors using the k-add mixture proposal distribution. The results were derived from an PMC sample of the field test data set.

Chapter 6

Discussion and Recommendations

6.1 Discussion

We began with the proposition that state estimation and data association are two distinct, but related, objectives of tracking that each require a different type of model in order to capture the relevant uncertainty. Too often this has been overlooked to the detriment of the application. It is important to be clear about what we hope to accomplish by tracking and what kind of uncertainty we care about in order to choose an approach suited to those needs. In the first chapter, we sought to clarify the relationship between the objective of tracking and the appropriate type of model, but it bears repeating here.

The first objective corresponds to tracking in the sense of following something. Tracking to reduce positional uncertainty is the canonical example. Given a sequence of noisy measurements of a target's position, the tracking algorithm integrates that information using models that describe the measurement process and the motion of the target to produce estimates of its positional state with (hopefully) less uncertainty than the raw measurements. Of course, when there are multiple targets or there is the possibility of receiving non-target measurements, it is no longer clear which measurements to use. In this case, data association is needed to determine the origin of each measurement so that the state estimates are not corrupted by information from the wrong source. The role of data association here is to maintain the state estimation process, which includes starting and stopping it as new targets appear and existing targets disappear. The tracking algorithm may seek the single best solution to the data association problem or it may use a probabilistically weighted average of multiple association hypotheses. Either way, any uncertainty with respect to the mapping of measurements to targets is effectively squeezed out to produce a collection of state estimates that represent just one interpretation of the data. This is true regardless of how principled the approach to dealing with the uncertainty is. For applications in which this uncertainty is either insignificant or deemed to be unimportant, such an approach is appropriate and one can draw on the numerous algorithms of this type in the literature.

There are, however, applications such as surveillance in which the uncertainty with respect to the mapping of measurements to targets is important and should be quantified for analysis or visualization. These applications require the ability to identify ambiguities in the interpretation of the data and compare different interpretations using

a well-defined measure of confidence. This is possible when the output of the tracker represents not target state distributions but rather the distribution of data association hypotheses. Such a tracker satisfies the second objective, which corresponds to tracking in the sense of finding something. Tracking to "connect the dots" in order to determine if any targets have been observed is the canonical example. While the detector ideally responds only to the presence of actual targets, in practice there will be false alarms. And even if the detector never produces a false alarm, we are still left with the association problem. To address these issues, the tracking algorithm processes a set of measurements and identifies the sequences that appear to have originated from the same target. This is equivalent to recognizing the time-varying patterns of the underlying target motion. The resulting mapping of measurements to targets tells us what has been observed and measurements not associated with any target can be rejected as false alarms. Of course, these decisions are usually informed by target state estimates made internally by the algorithm. Those estimates are not reflected in the output, however, recovering them for any target in the mapping is itself a well-defined state estimation problem since we know which measurements to process. Now, unlike the target state, whose uncertainty can usually be represented by known a probability distribution, the uncertainty of the mapping does not admit a closed-form representation. Thus, the output of the tracker consists of a sample drawn from the distribution of data association hypotheses from which application-specific quantities of interest can be estimated and the uncertainty of those estimates quantified.

Fox et al. [60] appear to be the only ones to have pointed out that having such a sample enables us to do more than just produce a single collection of state estimates that approximates the Bayes' optimal solution (i.e. minimizes the mean squared error) or is based on the maximum a posteriori (MAP) mapping. They demonstrated the benefit of having a representation of the distribution of data association hypotheses by identifying ambiguities in the target crossing patterns. They also noted the importance of considering multiple hypotheses in the face of such ambiguities. One of the contributions of our work is to make it clear that when the primary objective of tracking is to understand what has been observed, the ability to characterize this kind of uncertainty is critical. Of course, after analyzing the distribution of data association hypotheses, we may very well be interested in using some particular mappings, or even subsets of those mappings, to generate state estimates. Obtaining these estimates is then just a

series of well-defined state estimation problems.

While the literature does contain tracking algorithms that solve the data association problem directly, they are relatively rare compared to those that treat tracking as a state estimation problem. Only the Monte Carlo-based data association algorithms discussed in section 1.2.2 produce output suitable for the second objective. This is due in large part to the fact that the tracking field was born out of a need to meet the first objective. Consequently, the view of data association as a subproblem of state estimation is well entrenched and most of the research on data association has focused on the problem from this perspective. This matters for two reasons. First, despite their theoretical advantages, distributional models of the data association problem have received little attention because they are not required for state estimation; a point estimate of the mapping is sufficient and usually easier to compute. This is exacerbated by the nature of the problem itself. Data association is too difficult to solve optimally in the general case so we are forced to rely on approximations, which are often tailored to the role of maintaining the state estimation process. Second, even the existing algorithms that produce a sample drawn from the distribution of data association hypotheses are not widely recognized as providing a unique and independently useful information. As a result of all this, the options available to the practitioner are limited and the even those have not been well-studied, especially with respect to the quality of the sample they produce. Our work sought to change that by developing a new data association model that extends the scope and flexibility of existing models. In addition, we proposed two inference algorithms, based on different theoretical foundations and with different potentials for effective implementation. A thorough evaluation was then conducted for different configurations of the model and inference algorithms in the context of wide-area surveillance of human targets using radar. The model configurations consisted of numerous prior distributions over cluster labels and a variety of target motion models. The configurations for each inference algorithm consisted of various sampler proposal distributions.

In chapter 2, we presented the associative tracking model designed to satisfy the data association objective. It is based on a generative model of the posterior distribution over data association hypotheses, which is of the form:

$$p(\omega \mid \mathbf{Y}) \propto p(\mathbf{Y} \mid \omega)\, p(\omega)\,.$$

The model shares the same basic form as those used in existing multiple-scan Monte Carlo data association algorithms. This includes defining the target likelihood based on the joint measurement and state distribution with the state then marginalized out. However, our model differs from the existing work in three important ways.

First, we augmented the space of measurements to include a discrete element that indicates the absence of a detection. This enabled us to incorporate a state-dependent probability of detection (P_D) into the target likelihood model. Such support has traditionally been limited to algorithms that process the raw sensor data rather than discrete detections. The "negative" measurements (i.e. those corresponding to no detection where one was expected) are inferred from gaps in a sequence of target measurements, which are known from the conditional mapping. Exploiting the so-called "negative information" conveyed by these missed detections has been shown to improve performance in tracking applications. In traditional algorithms, this usually takes the form of reasoning about occlusions that is incorporated into track scores or assignment cost matrices. For distributional data association models that process detections, taking advantage of negative information is more difficult. Previous work [170, 171, 135] has recognized that in most applications P_D is not equal to one and thus should be incorporated into the model. The solution there was to assume a constant P_D, the effect of which was captured by the prior, $p(\omega)$. Unfortunately, this assumption of a uniform detection process does not allow one to model the effects of terrain obscuration or sensor gaps, which are often a problem in ground target tracking. Even in situations where the detection process cannot be modeled precisely and P_D can reasonably be regarded as constant in regions with sensor coverage, an effective compromise is to use a binary P_D model so that regions where no measurements are possible can be assigned a near-zero probability of detection.

Our approach provides the framework necessary to incorporate either precisely defined P_D models or simple occlusion-based P_D models into the target likelihood. However, this flexibility comes with a caveat. The form of the model, $P_D(\mathbf{x})$, will affect the form of the measurement model which will in turn affect whether the integration over the states in the target likelihood can be computed in closed-form. For performance reasons, we generally want to retain this property, but the only forms that will preserve it are a Gaussian function or a constant. Since neither of these are compelling, we can instead use a Gaussian approximation to the joint measurement and state distribution,

based on an arbitrary P_D model. We can also generate local constant P_D approximations on the fly during the integration process. These are computed at each time step in the lifetime of every target based on the expected probability of detection given the current state estimate.

The second way that our model differs from the existing work is in how the mapping of measurements to targets is defined. Typically, ω is defined using a set of indicator variables that identify the origin of each measurement. The values of these indicators can be thought of as cluster labels, where one cluster represents the false alarms and each of the other clusters represents a target. We have introduced a second set of indicator variables that identify whether the corresponding measurement is a false alarm. Together, these two sets of indicator variables, denoted by Z and $Z0$, comprise ω in our model. Although seemingly redundant, this notational change enabled us to factor the prior into separate models for the false alarm and target clusters. This is important because, despite the fact that conceptually each cluster in Z just represents a different measurement origin, the false alarm cluster is distinct from the others. While the size of any given target cluster is a priori unknown, we actually have strong beliefs about the size of the false alarm cluster. We even have a good idea of how the number of false alarm measurements is expected to be distributed across scans. Thus, it makes sense to represent these beliefs separately from those regarding the target clusters, which can be described using the same model. Not only does this improve the clarity of the data association model, it makes it possible to work directly with broader class of cluster label priors.

Finally, our model treats the false alarm rate as a random variable with its own hyperprior distribution. We assumed that the occurrence of false alarm measurements in any application could be modeled and that that information should be reflected in the prior, $p(\omega)$. However, we did not expect that the usual model based on a Poisson distribution with a known constant rate parameter would always be adequate. So, we extended the model by placing a gamma hyperprior on the rate parameter to characterize its uncertainty. This gives us a more flexible model for the distribution of false alarm counts, one where the variance need not be equal to the mean, and allows us to infer the arrival rate along with ω.

To sample from the associative tracking model, we proposed two algorithms based on different Monte Carlo methods, each with a different theoretical basis for its

validity. In addition, for each algorithm we considered multiple implementations of the proposal distribution. The reason is that efficient sampling from a high-dimensional distribution, such as $p(\omega \mid \mathbf{Y})$, is still very much an open problem and we have sought to determine what might be effective in this case. Unfortunately, none of the techniques commonly used to improve the sampling performance in high-dimensional models are useful for associative tracking. Since ω is discrete, Hamiltonian (or hybrid) Monte Carlo is not directly applicable and it is not clear how dimensionality reduction could be applied to the space of false alarm and target cluster indicator variables without losing any critical information. In previous work on multiple-scan distributional data association models, two types of sampling algorithm were proposed but neither of them proved to be effective on our data. The Gibbs sampler used by Fox et al. [60] was appealing because it implies a specific choice for the proposal distribution. However, even with the recommended switch step to improve the performance, we found it mixed very slowly. The Markov chain Monte Carlo (MCMC) samplers used by Oh et al. [135] and Yu et al. [199, 201] performed somewhat better because of their ability to modify multiple indicator variables in unison, but still mixed slowly and suffered from low acceptance rates, with the add moves being frequently rejected. This left us to consider different Monte Carlo algorithms and various proposal distributions.

The first algorithm was based on MCMC with a proposal distribution consisting of a mixture of reversible moves that modify the cluster assignments in various ways, similar to what has been done before. Where we deviated from those approaches is in how the moves themselves were implemented. Our goal was to address deficiencies with the add, extend, and update moves, which were not selecting favorable sequences of false alarm measurements to become target measurements with high enough probability. Oh et al. dealt with the issue by preprocessing the measurements to construct a neighborhood tree in which the edges indicate that the connected measurements can be plausibly associated based on a simple distance criterion. When reassigning false alarm measurements, the selections were then constrained to follow the edges in the tree in order to avoid making proposals with negligible likelihood. Yu et al. suggested using data-driven proposals to increase their likelihood but only applied them to appearance-related parameters and not to the cluster assignments. Our work extends the neighborhood tree concept into what we can the feasible graph and then uses it to make data-driven proposals for ω. In the feasible graph, each edge represents a plausible association of the

corresponding measurements based on an arbitrary model for a track consisting of just two measurements, which we call a tracklet. The tracklet model can take any form deemed to be appropriate, including that of a discriminative model. It need only return the likelihood of the association which is taken to be the weight of the corresponding edge. The graph was then used to approximate the joint likelihood, $p(\mathbf{Y} \mid \omega)$, when evaluating the probability of proposal candidates. For the add, extend, and update moves, we exploited the directed acyclic structure of the graph to efficiently find minimum cost paths, where the cost is an approximation of the negative posterior probability, that are likely to correspond to tracks.

Two different proposal distributions were considered for the MCMC sampler; both based on the same kernel mixture structure. We have termed them the default and alternative. The only difference between the two was that the alternative allowed individual moves to propose remaining in the present state. In the default, a dummy move was added to the mixture proposal provide an explicit means of keeping the status quo. The argument for the alternative is to reduce the tendency of the sampler to make transitions that will promptly need to be reversed when the wrong type of move is selected (e.g. an add move when the current state contains the correct number of clusters). As it turned out, the performance differed substantially depending on the approach, but the question of which was superior could not be resolved during our preliminary investigation. So, we added the handling self-transitions in the proposal distribution to the set variables to be studied in this work. The literature offers no guidance on this issue because there is no example of the current state being explicitly considered among the proposal candidates nor is there any discussion of why this should be the case. However, it should be noted that Oh et al. did provide an explicit option to keep the status quo fifty percent of the time in their single-scan algorithm.

MCMC is a widely used method of sampling from complex, high-dimensional probability distributions. Its primary appeal lies in the fact that it is relatively easy to construct a chain with the required invariant distribution. Unfortunately, the only thing that this guarantees is that in the limit of infinite sample size, the empirical averages used to estimate the expectations of functions with respect to this distribution will converge. In practice, the sample size is not only finite but often severely limited by the available time and computing power. How accurate these estimates will be for a given sample size is impossible to determine in general. However, the rate at which the

empirical averages converge is related to the mixing speed of the chain which in turn relates to how quickly the sampler explores the support of the distribution. Thus, we know in principle how to maximize the chances of generating a sample that represents the important features of the distribution after a fixed number of iterations. Accordingly, we have put considerable effort into the design of the proposal distribution in order to encourage rapid mixing. With the add move in particular, great pains were taken to ensure that large clusters could be created in a single step. Yet, despite all of this, questions about the convergence and quality of the estimates remain. This motivated us to consider a second sampling algorithm: population Monte Carlo (PMC). Our work is the first to apply this algorithm to the data association problem.

PMC is an iterated importance sampling scheme designed to generate a fixed-size sample from an arbitrary probability distribution. Unlike MCMC, which iteratively generates an approximate sample, PMC repeatedly simulates the entire sample. Its appeal is twofold. First, the approximation to the target distribution is valid at every iteration and thus does not require convergence time. Second, PMC is amenable to adapting the proposal distribution since ergodicity is not an issue. In fact, it is possible to use different sample-dependent proposal distributions at every iteration and for every element (i.e. particle). Estimates of expectations based on the sample remain unbiased (or approximately unbiased when the target distribution is unscaled) even when the proposals depend on past performance. A practical benefit is that PMC is also amenable to parallel computation.

Three different proposal distributions were considered for the PMC sampler; all three based on a mixture of reversible moves. Adaptation of the proposals was performed using the Rao-Blackwellized D-kernel PMC algorithm, which adjusts the mixture weights after each iteration based on the sample. The first two proposal distributions were the same ones that we used with the MCMC sampler in order to provide a means of comparing the two sampling algorithms under similar assumptions. The third proposal distribution was based on a mixture of k-add moves, where each one modifies ω such that the number of clusters is equal to the zero-based index of the move within the mixture. It represents an attempt to encourage a diverse sample by allowing each proposed set of indicators to have any number of clusters, up to some maximum value determined by the number of mixture components, regardless of the initial state. Since the expected number of clusters is assumed to be unknown, we rely on the adaptation

to learn the appropriate mixture weights.

In chapter 3, we took up the question of how to define the prior distribution of target cluster labels, $p(Z \mid \cdot)$. This decision is critical to the performance of the model because the likelihood component by itself will tend to favor mappings of measurements to targets in which the true tracks are split into fragments that each fare well under the motion model. Thus, a prior is needed that will encourage the continuity of those tracks and discourage short spurious tracks. Given the fact that we generally know nothing about the correct number of clusters or their sizes, it is far from clear what the best choice would be. The priors used by existing distributional data association models reflect two basic philosophies. The first assumes that we actually have relatively specific beliefs about the number and sizes of the clusters. These beliefs are represented by models of the appearance and disappearance of targets. The second assumes that we have no such knowledge and instead expresses a general belief that larger clusters are more plausible. Our view is that vague assertions, such as the one embodied by the second philosophy, are preferable for associative tracking since the premise is that we do not know what has been observed.

Despite the fact that objective cluster label priors have been proposed before, their performance in tracking applications remains poorly understood. In addition, none of the existing approaches have addressed the matter of how to set the parameters so that the distribution actually reflects our beliefs. Therefore, we decided to treat the prior distribution of target cluster labels as another variable to be studied and presented the candidates in their own chapter. This included those that have been used before, two cluster label priors that had not previously been applied to tracking, and two new nonparametric priors.

One of the cluster label priors from the literature is a property-based prior designed to express a preference for large clusters. We showed that by ignoring the combinatorial structure of the space of cluster labels, such priors obscure the relationship between their parameters and the statistics of the distribution, which makes it extremely difficult to determine suitable values for the parameters. A more principled approach with a similar preference, also in the tracking literature, is to use a nonparametric prior based on the Dirichlet process (DP). However, this was demonstrated without any explanation for how its critical tuning parameter, the concentration parameter, was set. Such priors have been ruled out by some researchers in other fields due to the sensitivity

of the results to the concentration parameter, so the matter cannot simply be ignored. We resolved this problem by placing a hyperprior on the concentration parameter that is defined based on the general belief that a small numbers of clusters are more plausible. This is possible because the relationship between the concentration parameter and the implied distribution of cluster counts is known.

Although the DP prior does reflect the kind of vague beliefs that we think are appropriate, it also implies a distribution of cluster sizes that heavily favors small clusters, which may not be desirable for tracking applications. Thus, we also considered a uniform process (UP) prior, which produces a flat distribution (ignoring the effects of the finite sample size) of cluster sizes. The UP prior was developed as an alternative to the DP prior for DNA motif clustering applications and our work is the first to apply it to target tracking. Unlike the DP, the UP does not induce an exchangeable distribution over the cluster labels and thus does not enjoy some of the theoretical properties that make the DP attractive. However, the UP prior has been shown to be approximately exchangeable and we showed how the calculation of $p(Z \mid \cdot)$ could be improved. The performance of the UP prior is also highly dependent on a concentration parameter that must be set appropriately. We handled this using a hyperprior defined analogous to the one used for the DP prior. The required relationship between the concentration parameter and the implied distribution of cluster counts was not known, so we derived a satisfactory approximation.

Based on the notion that it may be desirable for the distribution of cluster sizes to favor large clusters, we also introduced two forms of what we called the "egalitarian process" (EP). The term reflects the fact that the clustering process was defined so that at each step, the next assignment was most likely to be to small cluster. The idea was to express a preference for large clusters by encouraging each to be as large as possible. The two forms of the EP differ in the strength of the preference. Like the UP, these procedures fail to induce an exchangeable distribution over the cluster labels but are approximately exchangeable. We proposed a way to calculate $p(Z \mid \cdot)$ and derived an approximation to the distribution of cluster counts, as a function of the concentration parameter, needed to define the hyperprior.

The hierarchical uniform prior (HUP) was introduced in the statistics literature as an objective distribution of measurement partitions for clustering applications. It was developed as an alternative to both the DP and UP priors. The HUP is defined as the

product of a uniform distribution over partitions with a given the number of clusters and a distribution over the number of clusters that expresses a preference for a relatively small number of clusters. In other words, it treats all mappings of measurements to targets with the same number of clusters as equally likely. The UP prior does this as well, approximately, but the implied distribution of cluster counts is very different. For the HUP, that distribution can be specified directly. The original implementation suggested using either the Poisson-Jeffery's or Poisson-intrinsic prior, but we opted for a similarly shaped distribution with exponential-like decay with the rate specified by the likelihood ratio of the smallest and largest numbers of clusters. Our work is the first to apply the HUP to tracking problems.

Since the measurements are collected in scans and we assumed that each target will yield at most one measurement in a given scan (i.e. no split measurements), there are configurations of Z that should be assigned zero probability by the prior. Thus, the scan constraint effectively truncates $p(Z \mid \cdot)$, which has implications for how we determine the parameters in the property-based and nonparametric priors. Unfortunately, it is generally intractable to account for impact of the scan constraints exactly because the number of possible configurations is too large. For the nonparametric priors, we prosed a simple scheme for modifying the implied distribution of cluster counts to reflect constraints. While not exact, the approximation does improve the hyperprior considerably. In the case of the HUP, the truncation renders the distribution over partitions with a given the number of clusters non-uniform. To correct for this, we need a way to compute the scan-constrained number of configuration classes and the number of partitions in each configuration class. We proposed methods of approximating both of these values. The effect of the scan constraint on the property-based priors was not considered because the relationship of the parameters to $p(Z \mid \cdot)$ is generally unknown even without the constraint.

In chapter 4, we looked at how to tailor the associative tracking model to the motivating application: wide-area surveillance of human targets using radar. Specifically, we considered various means of specifying the dynamic model used to define the target likelihood. This particular combination of sensor and target has not been well-studied because the ability to reliably detect people at long range using radar is a relatively recent phenomenon. Tracking humans typically relies on appearance information to aid the data association process. However, we assumed, based on the current

state of technology and for the purpose of establishing a performance baseline, that no such information is available. Thus, we are relying on the dynamics for discriminating targets from spurious detections. This is complicated by the facts that humans are highly maneuverable and we are observing them on a coarse time-scale due to inherent difficulties in detecting such targets.

We began by studying the dynamics of human walkers from the perspective of the surveillance radar. This meant sampling the kinematic state with periods from 7.5 s to 60 s. The state was measured precisely using a differential global positioning system (DGPS) receiver worn by a test subject moving in three different types of terrain. The data collected was used to guide the model selection process and to calibrate the noise parameters. It was used for the performance evaluation conducted in the next chapter. These studies established two important facts. First, that the target speed is well-modeled in the discrete-time domain as a Gaussian. Second, that the target speed and heading are effectively uncorrelated after 30 s and 40 s, respectively. Consequently, we are justified in using efficient Gaussian models for the state transition distribution. We must also expect that models of correlated speed and heading are unlikely to provide significant benefit in practice.

The state of the target was defined to be its position and velocity. In all cases, the position was represented using 2-D Cartesian coordinates. Of the four basic dynamic models that we considered, one used a Cartesian representation of the velocity while the others used a polar representation. The polar representation was deemed preferable because of the convenience of modeling the speed and heading separately. Previous work on people tracking has used Cartesian velocity models exclusively, so we considered one as well. It was based on the standard assumption of a Wiener process model for each velocity component; the well-known continuous white noise acceleration (CWNA) model. However, we extended the model to optionally make predictions along a preferred heading. The preferred heading can be set to point to the current measurement when one wishes to model unconstrained maneuvers or it can be set based on local terrain or intent constraints. The motivation for this was the assumption that the maneuver time constant is much less than the sample period. This implies that between samples, a target can make significant changes in heading. Under these circumstances, the usual approach of increasing the process noise covariance to accommodate the maneuver is undesirable because it would require such a large increase that the measurements would

largely determine the filtered state estimate. By aligning the prediction with the preferred heading, the target speed model will have the desired influence on the estimated state.

All of the remaining dynamic models that we considered are based on nonlinear continuous-time models that relate the time derivatives of the target's speed and heading to the time derivatives of its Cartesian position. In order to ensure that the integrals in the likelihood calculations can be computed in closed-form, these models must be both linearized and discretized to put them into the required discrete-time linear-Gaussian form. The usual approach is to first linearize and then discretize the linear approximation. However, previous research has suggested that discretizing the system first and then generating a linear approximation can reduce the errors introduced by the linearization. For the sake of completeness, we derived the discrete-time linear models using both approaches although the latter is not entirely well-defined. At issue is how to compute the discrete-time process noise covariance for a nonlinear system. There is no accepted method for doing this and only crude approximations have been proposed. As an alternative, we introduced a novel approach based on the unscented transform (UT).

The first of the polar velocity dynamic models used independent Wiener processes to model the speed and heading. We also discussed a variant in which control inputs were used to reset the initial speed or heading state for each sample. The idea was to modify the speed or heading model to behave like a stationary Gaussian process centered on some preferred value given by the input. The second polar velocity dynamic model used independent Ornstein-Uhlenbeck (OU) processes to model exponentially correlated speed and heading. This required a small change to how the speed and heading states were represented. The speed and heading states were replaced in this model with the speed noise and heading noise states, which give the difference between the state and the steady-state value of the process. The steady-state value speed and heading are set using control inputs and can be interpreted as the target's preferred values. Such models are often used for vehicle and ship dynamics, but we appear to have been the first to apply them to human motion in a tracking application. Finally, the third polar velocity dynamic model combines the first with path-constrained state predictions. For this purpose, we introduced a novel implementation of a soft constraint designed to express a preference for path following, but without assuming that target

moves along the centerline or strictly parallel to the path. The idea was for the path to be a proxy for a trajectory through favorable terrain.

In chapter 5, we evaluated the performance of both the associative tracking model and the target dynamic models in the context of the motivating application. We began by testing the consistency and accuracy of the state estimates computed using the proposed dynamic models. The goal was to determine viability of each model and to select the best one. We also sought to quantify the reduction in positional uncertainty that state estimation can provide in realistic surveillance scenarios. It turned out that the basic CWNA model worked reasonably well, yielding position estimates that were slightly biased (approximately 0.14 m, on average) but with well-matched variances. The speed estimates, however, were strongly biased (approximately 0.16 m/s, on average) with overstated variances. Making the predictions along a preferred heading aligned with the current measurement improved the accuracy and consistency of the position estimates, but left the quality of the speed estimates largely unchanged. These position estimates tended to be closer to the true state as evidenced by the lower divergence metrics. Using the CWNA model, we also demonstrated the danger of using the normalized (state) estimation error squared (NEES) statistic to set the process noise as is often advised in the literature. In our example, this led to increased bias but improved variance matching. However, the resulting estimators actually led to an increase in the expected position error relative to the measurement error. When using the second configuration, the CWNA model reduced the expected position error by just under 9%, on average.

For the uncorrelated position–speed–heading (PSH) model, we considered various approaches to modeling the speed and heading. Using independent Wiener process models turned out to be problematic. While the consistency of the position estimates was similar to those from the CWNA model, they failed to improve the expected position error in almost all experiments. By applying control input to center the heading process on a preferred heading aligned with the current measurement fixed this problem. The innovations, however, were only approximately white with large numbers of slight outliers for small lags. Applying control input to center the speed process on a preferred value equal to the average target speed over the terrain had mixed results. When paired with a Wiener process heading model, the estimates diverged; the only example of divergence among all of our experiments. With the preferred heading model,

the estimates were reasonably consistent and with excellent accuracy. Using this configuration, the PSH model reduced the expected position error by just over 21%, on average.

For the exponentially correlated PSH model, we compared the different approaches to generating a discrete-time linear approximation of the system and found no significant performance differences. The linearized discretization approach did enjoy a slight advantage with respect to consistency and divergence from the true state, but the benefits were smaller than expected. It turned out that using the control inputs to configure the heading model such that the initial value aligned with the current state and the steady-state value aligned with the current measurement led to inferior results. This was surprising as it is arguably the most realistic heading model that we studied. Overall, using correlated speed and heading models offered no compelling advantage over their uncorrelated counterparts and yet required significantly more computation. The variance matching did improve, but the bias and accuracy was comparable. The best performing configuration of the exponentially correlated PSH model reduced the expected position error by approximately 19%, on average.

As expected, introducing the path constraints into the PSH model yielded the most accurate position estimates. In the mountain terrain, the expected position error was between 7 m and 12 m, depending the range of the target. Even in flat terrain at near range, the lowest error was still 5.3 m. These values represent about a 2–3 m improvement over the unconstrained models. The accuracy of the speed estimates, on the other hand, was actually slightly lower given the path constraints. Applying control input to center the speed process on a preferred value equal to the average target speed over the terrain improved the accuracy of all estimates compared to using a Wiener process speed model. However, this slightly worsened the already moderate consistency of the estimates by overstating the variance to a larger degree. Using this configuration, the path-constrained PSH model reduced the expected position error by approximately 34%, on average. This gain is based on the assumptions that data association is performed correctly, there are no missed detections, and that we have an accurate model of the target's trajectory. Thus, it represents the upper limit of what we can expect from state estimation for the given combination of target and sensor.

Next, we looked specifically at the feasible graph, which plays a vital role in the inference process by enabling the data-driven proposals and truncating the support

over which the space of associations is searched to eliminate regions of negligible probability. We proposed the following means of managing the density of the graph during its construction: applying a conservative maximum speed gate, enforcing a maximum association time, and limiting the out-degree of every vertex to the n-best associations based on the tracklet probability. Our experiments confirmed that under realistic conditions, the resulting graph can reasonably be expected to retain almost all of the edges between vertices corresponding to consecutive measurements of the target. We have termed these degree-one edges, where the degree indicates the order of the difference in the sequence of successive target detections. Even in the challenging case of $T = 60\,\mathrm{s}$ with a non-uniform binary P_D model, at least 89% of the degree-one edges were present in the graph. The primary reason for the loss of these edges was too many missed detections. This can be mitigated, at the expense of a denser graph, by increasing the maximum association time. However, any acceptable association time should be justified by the model. In the event that degree-one edges are missed because of model mismatch, the high-degree edges provide some redundancy that can enable us to still retain a significant fraction of a given track. Of course, as the degree increases, the percentage of those edges present in the graph will tend to decrease sharply. In practice, keeping the ten best edges worked well. Whether the maximum out-degree can be reduced further without severely impacting the inference depends on the tracklet model. In these experiments, except for the discriminative tracklet model, degree-one edges were assigned the largest weights in all but a handful of cases so we could reduce the graph density further without huge losses.

Two types of tracklet model were studied to gauge their effectiveness. The first was a generative model that used the likelihood components of the associative tracking model itself to define the tracklet probability. Thus, we had an instance of this tracklet model for each underlying dynamic model. The second was a discriminative model defined based on the implied speed of a straight-line trajectory between the two measurements. We assessed the quality of each tracklet model by using it to construct a feasible graph and then analyzing the edge weights to determine how well it discriminated between true associations and erroneous associations. The discriminative power was quantified using the H-measure, which describes whether a randomly chosen true associations in the graph is likely to have higher tracklet probability than an erroneous association. This has implications for both the construction of the feasible graph and

the quality of the data-driven proposal distributions.

With respect to the model type, generative was the clear winner. It turned out that the discriminative tracklet model performed relatively poorly because, while it tended to assign high probabilities to the true associations, many erroneous associations were also assigned high probabilities. However, unlike the generative models, the discriminative model actually performed better as the sample period increased, which might be beneficial for some applications. At any rate, we believe that the true advantage of using a discriminative tracklet model will come when incorporating high-level information that would be infeasible to express in a tractable generative model. Until then, it appears that the best strategy is to use generative formulations when possible. In the future, we might consider a discriminative model constructed by taking the conjunction of a generative likelihood model to effectively score the local kinematic smoothness of the track and a global track score to indicate how likely it is that the track exhibits the desired high-level behavior such as goal-seeking. Note that the prior term could still be included as a regularization term.

Regarding the performance of the different generative models, the results were not quite what we had expected. The experiments revealed that the discriminative power, as expressed by the H-measure, is not particularly well correlated with the accuracy of the underlying dynamic model. For example, the basic CWNA model actually outperformed a more accurate and more consistent PSH model, based on the preferred speed and heading, in the majority of the experiments. Despite being one of the weakest dynamic models in terms of its accuracy and consistency, the basic CWNA model tended to exhibit the most discriminative power among all models for short sample periods. For long sample periods, however, the path-constrained models dominated as expected. It seems that the implicit preference for smoothness induced by the Wiener process models was beneficial in terms of discrimination. The results of these experiments suggest that the optimal dynamic model for the associative tracking model (i.e. tracklets and likelihood) will not necessarily be the same as the optimal model for a filtering pass on the sampler output intended to produce accurate state estimates.

We also examined at the discriminative power of the tracklet models by comparing, for a range of lengths, the likelihood ratio between a true track and a sequence of invalid associations. The likelihoods were approximated using the tracklet probabilities by simply taking their product. For the generative tracklet models, we could

then compare these results to a baseline obtained using the true likelihood model. For all the models evaluated, the ratios grew with respect to the track length as expected. Among the generative tracklets, the agreement with the baseline values varied with the underlying dynamic model. Those based on Wiener process models tended to show less gain compared to the baseline. Tracklets based on dynamic models that utilized the preferred speed and heading tended to agree well with the baseline values, presumably because the dependencies between the measurements in the joint likelihood were weaker. For the discriminative tracklet model, there is no baseline but in absolute terms (i.e. the value of the likelihood ratio) and in terms of the growth rate the model showed good discrimination; better in fact than many of the generative models.

Finally, we looked at the quality of the posterior inferences obtained from the associative tracking model using two sets of simulated measurements and one set of field test data. For each data set, we evaluated the sample as a function of the sampler implementation and the cluster label prior used to complete the model specification. The sampler implementations that we considered consisted of the MCMC algorithm using the default and alternative proposal distributions, and the PMC algorithm using the default, alternative, and k-add mixture proposals. The prior models that we considered consisted of all those discussed in chapter 3. For the nonparametric priors, each was included in its unconstrained and scan-constrained form, as well as with a small fixed value for the concentration parameter. Three different property-based priors were considered including that from the seminal work of Oh et al. [135]. Since the "Oh" prior assumes knowledge of the target birth rate and disappearance probability, which we do not generally have, we introduced a hyperprior to model the uncertainty in these parameters. To make our assessments, we relied primarily on the simulated measurements because they naturally come with complete ground truth. For evaluating data association performance in wide-area surveillance applications, having complete ground truth is critical because it is not possible to reliably determine the origin of each real measurement and we can rarely be certain if any unscripted targets were missed in a large region. However, unlike traditional approaches to tracking simulators, we used precisely recorded true-target trajectories and limited the simulation to the just the measurement process and the false alarms; both of which should be well-characterized in any tracking application.

Regarding the choice of the prior, there was no clear winner as none proved to

be dominant. However, given an appropriate value of the concentration parameter, the DP prior was the most consistently strong performer. Our concern that the unequal distribution of cluster sizes favored by the DP would lead to an increased number of short spurious tracks turned out to be unfounded. For all of the nonparametric priors, the key determinant of the number of spurious tracks was not the properties of the clustering process but rather the value of the concentration parameter. When that value was too large, the number of spurious tracks was high for all of the priors, but their sizes were indeed distributed as the properties of underlying process would suggest. The strong preference for large clusters implied by the "rich-get-richer" behavior of the DP actually resisted the fragmentation of tracks. Despite the lure of its objectivity, the performance of models based on the UP prior tended to be weaker than the other nonparametric priors because it could not always keep the tracks intact. The EP and EP2 priors were also somewhat more susceptible to fragmented tracks. Note that between the two forms of the "egalitarian process", the EP2 enjoyed a slight advantage in terms of its expected scores and the quality of the resulting estimates. Although the preference for larger clusters expressed by the EP2 prior were better suited to the tracking problem than those expressed by the UP prior, compared to the DP prior there was no advantage.

For the nonparametric priors, the experiments confirmed the sensitivity of the results to the value of the concentration parameter, α. They also demonstrated that using the properties of the unconstrained process to determine this value is not effective. As we argued in chapter 3, the scan constraints effectively push the optimal value toward zero. Unfortunately, our attempt to account for the impact of the constraints in the hyperprior by approximating the relationship to the expected distribution over cluster counts was only partially successful. While it did lead to a lower expected α and improved results, the performance was inferior to that of a small fixed value, which indicates that the hyperprior is not dense enough for small values and that a different formulation is needed. In the next section, we will outline what that might look like.

Since the HUP prior no longer expressed the intended beliefs due to impact of the scan constraints, its performance was inconsistent at best. In some cases, it fared better than expected under the circumstances but generally lead to gross overestimates of the number of targets and poor expected scores. The scan-constrained version performed much better in comparison and actually performed well with data for which the likelihood component of the model was dominant. This suggests that the approxima-

tions to the configuration class counts and partition counts used in the implementation are reasonable in practice. However, we must bear in mind that there is no guarantee that the prior is expressing the intended beliefs. Models based on the scan-constrained HUP prior were prone to fragmenting the tracks and the prior did a relatively poor job of suppressing spurious tracks. These results suggest that objectivity is less desirable in the prior distribution over cluster labels than a well-calibrated preference for large clusters. Recall that the scan-constrained HUP prior already expresses a strong preference for small numbers of cluster but, for any given number of clusters, regards each configuration of the cluster labels as equally likely. Based on our experiments, the nonparametric priors are to be preferred. Note that the idea of explicitly considering the structure of the clustering problem may prove to be a useful means constructing new non-uniform priors in the future.

For the property-based priors, the "K" prior that naively expressed a preference for small numbers of targets performed poorly as expected and simply illustrated the potential cost of poor choice. The "Prop." prior expressed a similar preference based on the same parameterization, only stronger owing to the construction of the prior, plus a preference for associations between measurements close in time. However, despite its design, the "Prop." prior tended to perform poorly as well. Adjusting its parameters to strengthen the preference for small cluster counts might have helped, but without any sense of how those parameters relate to the properties of the implied distribution over cluster counts, it is not clear how to select a reasonable value or even how to model the uncertainty.

The performance of the "Oh" prior was inconsistent at best. Although the output was satisfactory in some experiments, models based on the "Oh" prior significantly overestimated the number of targets in general and were prone to producing fragmented tracks. Unfortunately, there is no easy remedy in terms of altering the parameters by improving the hyperprior. Unlike the nonparametric priors, which have a single parameter that controls the intensity of their preferences, there are two different parameters here whose interaction implicitly defines the preferences for the number of clusters and the distribution over cluster sizes. We have already tried to model the uncertainty in a way that favors a low target birth rate and remains largely agnostic with respect to the target disappearance probability. For the nonparametric priors, it is clear that we need encourage smaller α values; the critical question is how much. In the case of the "Oh"

prior, it is not clear that driving down the birth rate will help.

Regarding the implementation of the sampling algorithm, the results were mixed. Ignoring for a moment the choice of proposal distribution, the MCMC algorithm tended to generate samples from which plausible variance estimates could be made. For example, after processing a noisy data set that contained five targets, the number of targets in one experiment was estimated to be 5.2 with a standard deviation of 0.75. This is the kind of uncertainty in the distribution over data association hypotheses that we set out to model. Given the high false alarm rate, it makes a lot of sense that there would be some ambiguity about whether there are four, five, or six targets in the region. For that same experiment, however, the PMC algorithm generated a sample from which the number of targets was estimated to be 3.0 with a negligible standard deviation. Not only is this a poor estimate, but it indicates total certainty! In this example, the estimate derived from the PMC sample was less accurate, but in most experiments that was not the case. Overall, the estimates derived from the PMC sample tended to be more accurate than their MCMC-derived counterparts but with implausible certainty; almost all of variance estimates were negligible. The problem was the degeneracy of the sample weights, which occurred for all of the proposal distributions that we considered. However, given the quality of the estimates (uncertainty aside) and the flexibility to adapt the proposal distribution that the PMC algorithm provides, we still believe it is worth trying again in the future.

When using the default proposal distribution, which did not allow any move to include the current state among the proposal candidates, the resulting samples tended to overstate the number of targets. With the MCMC-generated samples, the number of targets was often significantly overestimated, although this was highly dependent on the prior, with some leading to reasonably accurate estimates. With the PMC-generated samples, we saw the same trend but the degree to which the number of targets was overestimated was generally much smaller. For both algorithms, switching to the alternative proposal distribution reduced the tendency to overstate the number of targets and improved the quality of the samples as measured by the expected combined score, which indicates how well Z matches the truth. Looking again at the number of targets, this brought excessively high estimates closer to the truth but also led to slight underestimates in many cases. The underestimates were caused by the failure to identify weakly resolved targets that were partially resolved in the previous samples.

In terms of the expected combined score, the samples based on the alternative proposal distribution were almost universally better, regardless of the algorithm. However, there is a caveat. The samples based on the alternative proposal distribution almost universally contained lower MAP probabilities. This indicates that with this proposal distribution the sampler is not exploring as much of the space, yet it seems to be to our benefit. It appears that while the sampler does not move as freely, the proposal distribution is good enough that those regions it does explore are relevant and lead to reasonable estimates. However, a consequence of the sampler exploring less of the space is that variance estimates shrink as a result of the truncation. Thus, the default proposal distribution is preferable in that regard. There is a case to be made for using either of these distributions. On one hand, the default proposal combined with the MCMC algorithm yields good variance estimates, but it is highly sensitive to the choice of prior. When the prior does not express sufficiently strong preferences, the risk of getting a sample that contains lots of spurious tracks is high. On the other hand, the alternative proposal seems to limit the potential impact of a mismatched prior at the expense of some sensitivity. When combined with the MCMC algorithm, an additional expense of this fail-safe behavior is less information about the uncertainty of the estimate.

Now, a bit more needs to be said about what "highly sensitive to the choice of prior" really means in practice because it reveals what is perhaps the most profound result from these experiments. That is, the realization that the MAP estimates were rarely the best ones. There were only two conditions under which they were the best, or close to the best. First, when the prior was well-matched to the problem in the sense that it expresses an appropriate preference and expresses it strongly enough. Second, when the sampler failed to thoroughly explore the space. To clarify those points, we can take the DP prior as an example. For the unconstrained DP prior and a given data set, the experiments with the MCMC sampler using the default proposal distribution yielded the highest MAP probabilities that were found among all of the sampler implementations. Those high probability configurations also happened to contain excessively large numbers of clusters. When using the alternative proposal distribution, the MCMC sampler failed to reach those high probability configurations and instead produced a sample whose MAP configuration contained a smaller number of clusters and was closer to the truth, but had a lower posterior probability. For the DP prior with

a small fixed α and that same data set, the MCMC sampler using the default proposal distribution yielded a MAP configuration that was very close to the truth. If we assume that the sampler continued to effectively explore the space in this case, then we can conclude that the prior with the strong preference for the smallest possible number of clusters led to a posterior model that is well-matched to the truth. Clearly, the first condition is the one that we ideally want to satisfy, but the alternative proposal distribution appears increase the chances that the second will be satisfied when the prior is not well-matched.

The k-add mixture proposal was meant to test the impact of being able to make extremely large changes to Z in a single move, regardless of the initial state. Since each move in the mixture generates proposals with a different number of clusters, it relied on the weight adaptation of the PMC algorithm to learn the correct distribution over cluster counts. In this regard, the proposal distribution performed well. Overall, the resulting samples had relatively high expected combined scores and tended not to overstate the number of targets. However, despite the potential for creating highly diverse samples, its samples suffered from weight degeneracy to a similar extent as those using the default and alternative proposal distributions.

Finally, let us comment on the performance of the field test data in particular. The results provided a clear example of how the alternative proposal distribution can be effective. Using the default proposal distribution, the number of targets was grossly overestimated in all experiments and the resulting samples contained numerous spurious tracks. Switching to the alternative proposal distribution led to significant improvements; so much so that almost half of the experiments in this case produced essentially perfect estimates. The results also showed that the demonstrably false assumption of spatially uniform false alarms was an adequate approximation. In addition, the results demonstrated that the proposed associative tracking algorithm is robust in to imperfect false alarm, measurement, and probability of detection models.

6.2 Concluding Remarks

This dissertation has presented a thorough analysis of the target tracking problem from the perspective of data association. It showed how tracking as a means of determining what has been observed is properly performed using what we termed an

associative tracking algorithm. Such algorithms produce as their output a sample from the distribution over data association hypotheses. The space of data association hypotheses is enormous and efficiently exploring it to generate a useful fixed-size sample presents a significant challenge. Of course, even if we can reliably obtain a good sample, the associative tracking model must still be a sufficiently accurate description of reality in order for the inferences to be meaningful.

Since the existing algorithms were not entirely suitable for our motivating application, wide-area surveillance of human walkers, we proposed a new associative tracking model and new methods of inference. The critical questions of the prior distribution over target indicator variables and the sampling algorithm were specifically addressed in our experimental evaluation using several challenging data sets. The application-specific question of how to model the target's state was also addressed. Ultimately, we sought to find an adequate model and an effective sampling algorithm. With respect to those goals, we can only claim to have been partially successful.

Regarding the model, our research showed that there are various ways to describe the time-varying kinematic state of a sparsely observed human walker with reasonable accuracy. It also showed that even with path-based constraints on the target's trajectory, the gain in localization accuracy that state estimation provides is relatively modest. The prior distribution over target indicator variables (i.e. cluster labels) proved to be absolutely critical to the accuracy of the model in terms of its agreement with the truth. Unfortunately, none of the priors that we considered led to a model that produced acceptable results in all of the experiments. While the nonparametric priors came close to achieving this goal, especially the Dirichlet process (DP) prior, it was only when an arbitrary small value of the all-important concentration parameter was used. The use of a small fixed concentration parameter value was intended solely as a reference, but the results clearly indicate the type of changes that are necessary for a suitable hyperprior.

Regarding the sampling algorithm, the MCMC sampler with our default implementation of the proposal distribution proved to be effective given an appropriate model. Moreover, the estimates tended to stabilize after the sample size reached 5000, which suggests that the sampler was exploring the space relatively efficiently. The experiments showed that the alternative implementation of the proposal distribution, in which each move included the current state among the proposal candidates, caused the

sampler to explore less of the space given the same number of iterations. However, it managed to visit regions of relatively high probability and produced samples from which reasonably accurate estimates could be derived. Given an appropriate model, this reduced somewhat the accuracy of the estimates from the resulting sample. However, when the model was mismatched, it actually improved the quality of the sample, often substantially. With all three of the proposal distributions that we considered, the PMC sampler suffered from weight degeneracy and produced samples that could best be described as providing reasonable point estimates since they failed to convey any appreciable uncertainty. Additional research is needed before this approach can be made viable. Despite the success of the MCMC sampler, a viable PMC sampler remains an attractive goal because it eliminates concerns about convergence of the estimates and would lend itself to efficient parallel implementation. Given the flexibility to adapt the proposal distribution that PMC affords, there is reason to be optimistic that it can be made to work.

6.3 Future Work

Although we have learned a lot about tracking as a data association problem, more work is necessary before these algorithms are suitable for use in practical applications. Here, we will outline some of the promising directions for future research.

6.3.1 Cluster Label Priors

Even a perfect sampler is of limited value if the model provides a poor description of reality and, as our results demonstrated, the prior distribution over cluster labels is critical in this regard. Thus, improvements in this area are important.

6.3.1.1 Concentration Parameter Hyperprior

The nonparametric priors, especially the Dirichlet process (DP) prior, proved to be effective only in the case of an appropriately chosen concentration parameter, α. This, rather than the preferences with respect to cluster sizes, was the primary factor in suppressing spurious tracks and producing the desired results.

Our attempt to model the uncertainty in α based on the agreement between our prior beliefs about the number of clusters and the distribution of cluster counts

produced by the process did not prove to be sufficient because the resulting α values tended to be too high. The hyperprior clearly did not contain enough probability density near zero. This is likely due to inaccuracies in the estimates of the mean and variance of the cluster counts given the scan constraints. And the prospects for improving those in a principled manner do not appear to be good.

We propose to consider a simpler approximation with a bias toward small values of α. A gamma distribution with the shape parameter set to 1.0 is one such approach. In this case, the rate parameter could be set in part by considering an upper bound on the value of α. To estimate the upper bound, we could take the value of α at which the expected number of clusters produced by the unconstrained process is equal to the maximum number of clusters based on our prior beliefs. This calculation of $E[k \mid \alpha, n]$ would be performed with n equal to the number of scans implied by *Z0*.

6.3.1.2 Scan-Constrained Hierarchical Uniform Prior

While not among the best performing priors due to its susceptibility to fragmented tracks, the concept of using the configuration class and partition counts in the model may have some utility in constructing new types of prior distributions. We can envision using this technique to construct non-uniform priors with specific properties that properly account for the combinatorial structure of the clustering problem. However, since these counts must be approximated in the case of the scan constraints, and we have no bounds for the accuracy of our approximations, it would be interesting to see if there are ways to compute these counts exactly. Generating functions seem like a good place to start.

6.3.1.3 Beta Process Priors

We made the assumption that there were no merged measurements, which meant that each measurement could be assigned to at most one cluster. This implied that Z described a partition of the measurements, for which nonparametric models such as the Dirichlet process were appropriate. However, for radar applications, the assumption of no merged measurements is a simplification that does not match reality and one that we would like to relax.

Since the nonparametric models performed well, it would be desirable to use similar models in the case of merged measurements. One way to do this would be to

use a beta process model to induce a nonparametric prior over infinite binary matrices that encode the assignment of measurements to tracks. The resulting distribution over infinite binary matrices can also be described by the Indian buffet process (IBP).

6.3.2 Modeling High-Level Target Behavior

To improve the model, additional constraints on the motion of the target beyond the simple kinematic constraints already studied are desirable. For intent-driven targets, such as people, those constraints could be derived from goal-seeking or other high-level behaviors. These constraints would be impossible to represent using Gaussian likelihood models. Even if we were to consider non-Gaussian likelihood models, it would still be unclear how the high-level behavior effects the data generating process. Thus, we should consider switching to a discriminative model of $p(Z,\, Z0,\, \lambda,\, \alpha \mid \tilde{Y},\, a_\lambda,\, b_\lambda)$, which would still allow us to work with the same inference algorithms. As a starting point for the discriminative modeling, we might consider simply adding a term to the current generative model that computes a global track score based on some high-level information. Ultimately, we believe that the most useful practical tracking algorithms will be those that can be tuned to watch for very specific types of behavior. In challenging environments, it may not be feasible to track all people and then attempt to classify the tracks based on their high-level behavior. That capability will likely have to be built into the tracking model itself.

6.3.3 Terrain Influence

In the current work, we used path constraints to approximate the influence of the terrain on a target's motion and this proved to be highly beneficial. The path represented favorable terrain and the target's trajectory was assumed to be influenced by it. Unfortunately, this approach is inherently limited to relatively sparse path networks. It will break down if we attempt to create paths for too many terrain contours. Thus, a different approach is necessary to make more robust use of terrain information. In particular, what is needed, is a way to infer plausible trajectories, between the target's current state and the next measurement, given the terrain. One way that this could be done is by representing the terrain as an artificial potential field [106, 134]. This approach might also lend itself to the incorporation of high-level target behavior because we could include the attractive force of goal regions in the potential field or even the

repulsive force of something that a target wishes to avoid. Realistic movement through terrain could also be modeled by applying the properties of a fluid simulation (gas diffusion) [20, 118].

6.3.4 Online Processing

The fact that the associative tracking algorithm performs batch processing of the measurements limits its practical use. Existing multiple-scan statistical data association models also share this limitation. Currently, the accepted solution is to use a sliding window with the sampler initialized based on the output of the previous run. Unfortunately, this is not entirely satisfactory because it fails to address the critical question of how to summarize the distant past. While the sampler might benefit from a good initial state, there is no influence from the time up the start of the current processing window. When the model incorporates a nonparametric prior, the sizes of any existing clusters might very well be relevant.

There are two approaches that we might consider for summarizing what was learned from past measurements.

First, let us imagine the feasible graph in trellis form. The distant past could be represented using a special set of nodes in a hypothetical scan that precedes the first scan in the processing window. Those special nodes would correspond not just to measurements, but would represent the entire history of sequences in the distant past that terminate at their respective measurements. Associated with each one would be a state estimate and expected length of the paths leading to it. Whenever the feasible graph is constructed, these special nodes would also be linked to nodes in the current window using the tracklet model, which would make use of the current state. The likelihood calculation of any path including a special node would use the initial state and the cluster label prior would make use of the expected length, if necessary. The nodes would then be updated each time the processing window advances. Any nodes whose last association was too far in time from the first scan in the window would then be removed.

A second approach would be to use a particle set to summarize the past trajectories in the manner of Särkkä et al. [171]. In fact, the idea would be to extend their approach to multiple scans. This too would require the use of a sliding window, but in a somewhat different manner. For a given window position, we would generate a sample

from the multiple-scan posterior, once for each particle. To properly account for the past trajectories represented by each particle when generating the sample, each trajectory would be summarized by its length and final state. Those summaries would then be utilized by the sampler using the hypothetical scan technique to modify the feasible graph as outlined in the previous approach. The summaries, of course, would not need to be re-computed from scratch each time with the proper bookkeeping. Now, when the particle set is updated, each trajectory would only be extended based on the sampled associations with measurements in the first scan of the window. In this way, both the past and the full window of scans would inform the update. The details of exactly how the particle set is updated given all of these samples would need to be worked out because there are different ways it could be done. Using such an approach, the window size would only need to be set to the maximum association time, which would be much easier to sample from. Because of the overlap due to the sliding window, we would have very good initial states for each sample. The collection of samples generated for each particle would also be available for applications to visualize the uncertainty over the horizon of the processing window.

6.3.5 Inference

6.3.5.1 Improved Population Monte Carlo

Despite the problem of severe weight degeneracy observed in these experiments, we strongly believe that population Monte Carlo (PMC) sampling is worth a second attempt. Its flexibility with respect to adapting the proposal distribution based on past samples give us reason to be optimistic and the potential for efficient parallel implementations remains compelling.

In addition, we should also look at the mixture weight adaptation. The original motivation behind the D-kernel PMC algorithm was kernel bandwidth selection and it is not clear if the same ideas are applicable to the types of kernels that we considered. For example, add moves are only advantageous when the current state contains too few clusters. Increasing the weight of these moves based of survival will only be helpful up to a point. It is quite possible that there is a better way to adjust the weights when using these dimension-jumping-type moves.

We might also be able to take advantage of the fact that proposals in the PMC

algorithm need not be reversible.

6.3.5.2 Alternatives

Sampling from the posterior is asymptotically exact but very hard to do efficiently. Could we use variational inference to sample from a suitable approximation of the posterior instead? At the time of writing, no amenable approximation scheme was found. However, variational Bayesian methods are an active area of research and it would be worth trying again in the future.

Appendix A

Gamma Approximation Using Moment and Mode Matching

Moment matching is an efficient method of approximating an unknown probability distribution using a distribution with a well-known form. Given certain statistics of the unknown distribution, the parameters of the well-known distribution are chosen such that these statistics are matched to the extent possible.

Here, we apply the method to derive the parameters of a gamma distribution given various combinations of the desired mean, variance, and mode.

Let $x \sim \Gamma(a, b)$, where $a > 0$ is the shape parameter and $b > 0$ is the rate parameter. The probability density function is:

$$f(x; a, b) = \frac{b^a}{\Gamma(a)} x^{a-1} \exp(-bx)$$

and the statistics of interest are:

$$\begin{aligned} \mathrm{E}[x] &= \frac{a}{b} \\ \mathrm{Var}[x] &= \frac{a}{b^2} \\ \mathrm{Mode}[x] &= \begin{cases} \frac{a-1}{b} & 1 < a \\ 0 & 0 < a \leq 1. \end{cases} \end{aligned}$$

Matching the Mean and Variance

Given the mean, e, and variance, v, we seek the parameters of a gamma distribution such that:

$$\begin{aligned} \mathrm{E}[x] &= \frac{a}{b} = e \\ \mathrm{Var}[x] &= \frac{a}{b^2} = v. \end{aligned}$$

For $e > 0$ and $v > 0$, the mean and variance can be matched exactly using:

$$\begin{aligned} a &= \frac{e^2}{v} \\ b &= \frac{e}{v}. \end{aligned}$$

Matching the Mode and Variance

Given the mode, m, and variance, v, we seek the parameters of a gamma distribution such that:

$$\begin{aligned}\text{Mode}\left[x\right] &= \frac{a-1}{b} = m \\ \text{Var}\left[x\right] &= \frac{a}{b^2} = v.\end{aligned}$$

For $m \geq 0$ and $v > 0$, the mode and variance can be matched exactly using:

$$\begin{aligned}a &= 1 + m\left(\frac{m+\sqrt{m^2+4v}}{2v}\right) \\ b &= \frac{m+\sqrt{m^2+4v}}{2v}.\end{aligned}$$

Certain applications, such as proposal distributions for Monte Carlo sampling, can benefit from a gamma approximation with a slightly larger variance. We can derive such an approximation in a somewhat principled manner by determining the shape and rate to match the given mode and a modified expression of the variance based on the mean of the squared deviations from the mode. These are given by:

$$\begin{aligned}a &= 1 + m\left(\frac{m+\sqrt{m^2+8v}}{2v}\right) \\ b &= \frac{m+\sqrt{m^2+8v}}{2v}.\end{aligned}$$

Matching the Mean and the Mode

Given the mean, e, and the mode, m, we seek the parameters of a gamma distribution such that:

$$\begin{aligned}\text{E}\left[x\right] &= \frac{a}{b} = e \\ \text{Mode}\left[x\right] &= \frac{a-1}{b} = m.\end{aligned}$$

For $m \geq 0$ and $e > m$, the mean and the mode can be matched exactly using:

$$\begin{aligned}a &= \frac{e}{e-m} \\ b &= \frac{1}{e-m}.\end{aligned}$$

Matching the Mean, Variance, and Mode

Given the mean, e, variance, v, and mode, m, we seek the parameters of a gamma distribution such that:

$$\begin{aligned}\mathrm{E}[x] &= \frac{a}{b} = e \\ \mathrm{Var}[x] &= \frac{a}{b^2} = v. \\ \mathrm{Mode}[x] &= \frac{a-1}{b} = m.\end{aligned}$$

In general, it is not possible to match all three statistics exactly, so instead we solve for the parameters that minimize the error objective function:

$$J(a, b) = w_e \left(e - \frac{a}{b}\right)^2 + w_v \left(v - \frac{a}{b^2}\right)^2 + w_m \left(m - \frac{a-1}{b}\right)^2,$$

where w_e, w_v, and w_m are non-negative weights applied the squared error in the mean, variance, and moment matching, respectively.

For $m \geq 0$, $e \geq m$, and $v > 0$, the best match can be found as follows.

Let

$$\begin{aligned}w_1 &= w_e^2 w_m \\ w_2 &= w_e w_m^2 \\ w_3 &= w_e^2 w_v \\ w_4 &= w_e w_v^2 \\ w_5 &= w_m w_v^2 \\ w_6 &= w_m^2 w_v \\ w_7 &= w_e w_m w_v \\ w_8 &= w_m^2 w_v^2 \\ w_9 &= w_m^3 w_v \\ w_{10} &= w_e^3 w_m \\ w_{11} &= w_e^2 w_v^2\end{aligned}$$

and

$$\begin{aligned}
C_0 &= -e\left(w_1 + w_2 + v\left(w_3 + w_7\right)\right) + m\left(w_1 + w_2 - v\left(w_6 + w_7\right)\right) \\
C_1 &= w_1 + w_2 + e^2 w_3 + m^2 w_6 + 2emw_7 - v\left(v\left(w_4 + w_5\right) + 2\left(w_6 + w_7\right)\right) \\
C_2 &= e\left(vw_4 + w_7\right) + m\left(vw_5 + 3w_6 + 2w_7\right) \\
C_3 &= 2\left(w_6 + w_7\right) \\
C_4 &= mw_5 \\
C_5 &= w_5.
\end{aligned}$$

Let r be a root of the quintic polynomial:

$$C_0 + C_1 x + C_2 x^2 + C_3 x^3 + C_4 x^4 + C_5 x^5,$$

such that $r \in \Re$ and $r > 0$.

The shape and rate parameters are then given by:

$$\begin{gathered}
a = \Big[r^4\left(mw_8 + ew_4 w_m\right) + \\
r^3\left(\left(m^2 + v\right)w_8 + w_9 + \left(em + v\right)w_4 w_m + w_2 w_v\right) + \\
r^2\left(m\left(v\left(w_4 w_m + w_8\right) + 2w_9\right) + \left(ew_1 + \left(e + 2m\right)w_2\right)w_v\right) + \\
r\left(-e^2\left(w_{10} - vw_{11}\right) - v^2 w_{11} + \left(m^2 - v\right)\left(vw_8 + w_9\right) + \right. \\
\left(em - v\right)\left(2vw_4 w_m - w_1 w_v\right) + \left(em - 2v\right)w_2 w_v\Big) + \\
\left(e^3 w_{10} + m\left(m^2 + v\right)w_9 + mw_e^2 w_m^2 + 2mw_e w_m^3 + \right. \; / \\
\left. + mw_m^2 + \left(3e^2 + v\right)mw_1 w_v + \left(3em + 2v\right)mw_2 w_v\right)\Big] \; / \\
r\left(\left(w_e + w_m\right)\left(w_2 + e^2 w_3 + v^2\left(w_4 + w_5\right) + \left(m^2 + 2v\right)w_6 + 2\left(em + v\right)w_7 + w_m^3\right)\right) + 1
\end{gathered}$$

and

$$b = \frac{1}{r},$$

such that $a > 0$. In the event that there is more than one positive real root which yields a positive shape parameter, the error objective should be used to select the best parameterization.

Note that although the desired mean and mode may be equal, when both are zero the matching may not yield valid parameters. Depending on the variance and weights, there may no positive real roots.

Note also that by placing zero weight on the appropriate error term, exact (to within numerical precision) matches for the cases above can be obtained.

Appendix B

Numerically Stable Evaluation of $p(k \mid \alpha, n)$ for the Dirichlet Process

On occasion, we may wish to evaluate the distribution over the number of clusters induced by the Dirichlet process. A prime example is when eliciting a prior for the concentration parameter, α, which controls the number of clusters.

From Antoniak [3], we find the distribution to be:

$$p(k \mid \alpha, n) = S_1(n, k)\, \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)}, \tag{B.1}$$

for $k = 1, \ldots, n$ and where k is the cluster size, n is the number of samples, $S_1(n, k)$ is the unsigned Stirling number of the first kind and $\Gamma(\cdot)$ is the Gamma function.

Unfortunately, numerical issues can arise when evaluating this distribution. The problem is that for even modest values of α and n, each term in (B.1) can grow to exceed to range of 64-bit floating point numbers. For example, $S_1(500, 75) = 1.639 \times 10^{1069}$ and $\Gamma(500 + 25) = 1.313 \times 10^{1199}$!

Here we describe a numerically stable approach to evaluating $p(k \mid \alpha, n)$.

Stirling Numbers of the First Kind

Unsigned Stirling numbers of the first kind, $S_1(n, k)$, arise as coefficients of the rising factorial [195]:

$$x\,(x+1)\cdots(x+n-1) = \sum_{k=0}^{n} S_1(n, k)\, x^k.$$

They count the number of permutations of n elements which contain k permutation cycles.

Signed Stirling numbers of the first kind, $s(n, k)$, arise as coefficients of the falling factorial:

$$x\,(x-1)\cdots(x-n+1) = \sum_{k=0}^{n} s(n, k)\, x^k.$$

The signed and unsigned numbers are related as:

$$S_1(n, k) = |s(n, k)| = (-1)^{n-k}\, s(n, k)\,.$$

Unsigned Stirling numbers of the first kind can be calculated using the following recurrence relation:

$$s(n, k) = s(n-1, k-1) - (n-1) \cdot s(n-1, k)\,,$$

for $1 \leq k \leq n$ and where:

$$\begin{aligned} s(n,\, k) &= 0 \quad \text{for } k > n \\ s(n,\, 0) &= 0 \quad \text{for } n > 0 \\ s(0,\, k) &= 0 \quad \text{for } k > 0 \\ s(n,\, n) &= 1. \end{aligned}$$

Signed Stirling numbers of the first kind can be calculated using the following recurrence relation:

$$S_1(n,\, k) = (n-1) \cdot S_1(n-1,\, k) + S_1(n-1,\, k-1)\,,$$

for $1 \leq k \leq n$ and where:

$$\begin{aligned} S_1(n,\, k) &= 0 \quad \text{for } k > n \\ S_1(n,\, 0) &= 0 \quad \text{for } 0 > n \\ S_1(0,\, k) &= 0 \quad \text{for } 0 > k \\ S_1(n,\, n) &= 1. \end{aligned}$$

There is also an explicit formula [194]:

$$s(n,\, k) = \frac{(2n-k)!}{(k-1)!} \sum_{i=0}^{n-k} \frac{1}{(n+i)\,(n-k-i)!\,(n-k+i)!} \sum_{j=0}^{i} \frac{(-1)^j\, j^{n-k+i}}{j!\,(i-j)!}.$$

Numerically Stable Evaluation

To avoid arithmetic overflow, we would like to be able to calculate (B.1) as:

$$p(k \mid \alpha,\, n) = \exp\left\{\log S_1(n,\, k) + k \log(\alpha) + \log\Gamma(\alpha) - \log\Gamma(\alpha+n)\right\}.$$

This requires the ability to compute the natural logarithms $\log S_1(n,k)$ and $\log\Gamma(x)$, without having to explicitly represent $S_1(n,k)$ and $\Gamma(x)$.

The natural logarithm of the Gamma function is readily available in tools such as MATLAB and libraries such as the GNU Scientific Library. However, the natural logarithm of a Stirling number of the first kind does not appear to have a well-known solution.

Here we introduce a novel approach to computing $\log S_1(n,k)$ that uses ratios of Stirling numbers of the first kind.

First, let us define the ratio of signed Stirling numbers of the first kind as:

$$r(n,\, k) = \frac{s(n+1,\, k)}{s(n,\, k)}.$$

From Cacoullos and Papageorgiou [22][1], we see that the ratio can be calculated using the following recurrence relation:

$$r(n,\, k) = \frac{[r(n-1,\, k) + n - 1]\, r(n-1,\, k-1)}{r(n-1,\, k)} - n,$$

for $1 \leq k \leq n$ and where:

$$\begin{aligned} r(n,\, 1) &= -n \\ r(n,\, n) &= -\frac{n\,(n+1)}{2}. \end{aligned}$$

We can also see from Cacoullos and Papageorgiou that:

$$\frac{s(n,\, k-1)}{s(n,\, k)} = r(n,\, k) + n.$$

Next, we refactor the recurrence for unsigned Stirling numbers of the first kind to get:

$$S_1(n,\, k) = (n-1) \cdot S_1(n-1,\, k) \left(1 + \frac{1}{(n-1)} \frac{S_1(n-1,\, k-1)}{S_1(n-1,\, k)}\right).$$

The logarithm can then be calculated using the following recurrence:

$$\log S_1(n,\, k) = \log\,(n-1) + \log S_1(n-1,\, k) + \log\left[1 + \frac{1}{(n-1)}\,|r(n-1,\, k) + n - 1|\right], \quad \text{(B.2)}$$

for $1 \leq k \leq n$ and where:

$$\log S_1(n,\, n) = 0.$$

An alternative approach to calculating (B.1) without the natural logarithm of Gamma function exploits the fact that:

$$\begin{aligned} p(k \mid \alpha,\, n) &= S_1(n,\, k)\, \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} \\ &= \frac{S_1(n,\, k)\, \alpha^k}{\alpha\,(\alpha+1)\cdots(\alpha+n-1)} \\ &= \frac{S_1(n,\, k)\, \alpha^k}{S_1(n,\, 1)\, \alpha + S_1(n,\, 2)\, \alpha^2 + \cdots S_1(n,\, n)\, \alpha^n} \\ &= \frac{1}{\sum_{i=1}^{n} \frac{S_1(n,\, i)}{S_1(n,\, k)} \alpha^{i-k}}. \end{aligned}$$

[1] Note that m and n, as used in [22], are equivalent to our k and n, respectively.

The recurrence given by (B.2) can be used to calculate $S_1(n,\, k)$ for $k = 1, \ldots, n$ and then each term in the summation above can be found using:

$$\exp\left\{\log S_1(n,\, i) - \log S_1(n,\, k) + (i - k)\log\left(\alpha\right)\right\}.$$

Appendix C

State Estimation Performance

		$\mathbf{err}_{\mu_x}$ (m)	$\mathbf{err}_{\mu_y}$ (m)	$\mathbf{err}_{\mu_s}$ (m/s)	$\mathbf{err}_{\sigma_x}$ (m)	$\mathbf{err}_{\sigma_y}$ (m)	$\mathbf{err}_{\sigma_s}$ (m/s)	$\mathbf{rej}_{\mu_x}$ (%)	$\mathbf{rej}_{\mu_y}$ (%)	$\mathbf{rej}_{\mu_s}$ (%)	$\mathbf{rej}_{\sigma_x}$ (%)	$\mathbf{rej}_{\sigma_y}$ (%)	$\mathbf{rej}_{\sigma_s}$ (%)
CWNA/C1													
3 km	$T=15$ s	0.05	-0.27	-0.10	-0.65	-0.64	-0.15	0.24	0.27	0.84	0.02	0.03	0.92
	$T=30$ s	0.03	-0.31	-0.01	-0.66	-0.64	-0.13	0.29	0.26	0.37	0.03	0.02	0.95
	$T=60$ s	0.07	-0.08	-0.04	-0.64	-0.66	-0.35	0.21	0.17	0.96	0.01	0.02	0.99
5 km	$T=15$ s	0.13	-0.44	-0.18	-1.06	-0.99	-0.18	0.26	0.29	0.93	0.02	0.02	0.97
	$T=30$ s	0.04	-0.43	-0.06	-1.05	-1.01	-0.21	0.23	0.24	0.82	0.02	0.02	0.97
	$T=60$ s	0.31	-0.10	-0.14	-1.00	-0.97	-0.50	0.10	0.13	0.94	0.01	0.01	0.99
7 km	$T=15$ s	0.07	-0.60	-0.23	-1.47	-1.43	-0.24	0.29	0.29	0.94	0.03	0.03	1.00
	$T=30$ s	0.09	-0.52	-0.13	-1.43	-1.43	-0.30	0.25	0.23	0.87	0.01	0.01	0.99
	$T=60$ s	0.28	-0.22	-0.33	-1.27	-1.34	-0.65	0.08	0.08	1.00	0.02	0.02	1.00
CWNA/C2													
3 km	$T=15$ s	0.02	-0.03	-0.15	0.03	-0.08	-0.14	0.04	0.06	0.98	0.01	0.01	0.85
	$T=30$ s	0.03	-0.06	-0.02	-0.25	-0.32	-0.12	0.04	0.03	0.42	0.02	0.01	0.95
	$T=60$ s	0.03	0.02	-0.04	-0.54	-0.58	-0.33	0.04	0.04	0.96	0.01	0.02	0.99
5 km	$T=15$ s	0.10	0.02	-0.29	0.43	0.29	-0.14	0.27	0.33	1.00	0.02	0.02	0.47
	$T=30$ s	0.09	-0.06	-0.09	-0.13	-0.24	-0.17	0.03	0.05	0.88	0.01	0.01	0.95
	$T=60$ s	0.19	0.03	-0.14	-0.81	-0.79	-0.47	0.01	0.00	0.94	0.01	0.00	0.99
7 km	$T=15$ s	0.08	-0.07	-0.40	0.84	0.74	-0.17	0.41	0.61	0.99	0.05	0.05	0.37
	$T=30$ s	0.08	-0.12	-0.18	0.10	-0.04	-0.22	0.15	0.18	0.95	0.01	0.01	0.97
	$T=60$ s	0.16	-0.07	-0.31	-0.99	-1.05	-0.61	0.01	0.00	1.00	0.01	0.01	1.00
CWNA/C3													
3 km	$T=15$ s	0.06	-0.02	-2.38	-0.47	-0.44	-1.87	0.17	0.18	1.00	0.01	0.01	1.00
	$T=30$ s	0.06	-0.01	-2.10	-0.50	-0.53	-1.91	0.05	0.05	1.00	0.01	0.01	0.99
	$T=60$ s	0.03	0.05	-3.79	-0.61	-0.63	-3.54	0.01	0.03	1.00	0.01	0.02	0.99
5 km	$T=15$ s	0.16	-0.02	-2.64	-0.90	-0.86	-1.88	0.20	0.30	1.00	0.00	0.00	1.00
	$T=30$ s	0.09	-0.06	-2.23	-0.75	-0.70	-1.87	0.12	0.17	1.00	0.01	0.01	0.99
	$T=60$ s	0.26	0.01	-3.86	-0.87	-0.82	-3.54	0.03	0.04	0.99	0.01	0.00	0.99
7 km	$T=15$ s	0.11	-0.11	-2.87	-1.38	-1.30	-1.92	0.20	0.29	1.00	0.00	0.01	1.00
	$T=30$ s	0.12	-0.09	-2.36	-1.13	-1.12	-1.85	0.14	0.24	1.00	0.01	0.00	1.00
	$T=60$ s	0.28	-0.10	-4.02	-1.07	-1.01	-3.61	0.04	0.08	1.00	0.02	0.02	1.00

Table C.1: Consistency summary statistics for the CWNA model applied to targets moving in flat terrain (T1).

		$\mathbf{err}_{\mu_x}$ (m)	$\mathbf{err}_{\mu_y}$ (m)	$\mathbf{err}_{\mu_s}$ (m/s)	$\mathbf{err}_{\sigma_x}$ (m)	$\mathbf{err}_{\sigma_y}$ (m)	$\mathbf{err}_{\sigma_s}$ (m/s)	$\mathbf{rej}_{\mu_x}$ (%)	$\mathbf{rej}_{\mu_y}$ (%)	$\mathbf{rej}_{\mu_s}$ (%)	$\mathbf{rej}_{\sigma_x}$ (%)	$\mathbf{rej}_{\sigma_y}$ (%)	$\mathbf{rej}_{\sigma_s}$ (%)
CWNA/C1													
3 km	$T=15$ s	-0.13	-0.06	-0.12	-0.65	-0.65	-0.17	0.51	0.42	0.85	0.01	0.02	0.91
	$T=30$ s	-0.20	-0.15	-0.01	-0.66	-0.63	-0.14	0.58	0.44	0.56	0.01	0.02	0.92
	$T=60$ s	-0.02	-0.04	0.00	-0.67	-0.65	-0.37	0.24	0.21	0.91	0.01	0.01	1.00
5 km	$T=15$ s	-0.24	-0.15	-0.16	-1.04	-1.01	-0.19	0.55	0.47	0.87	0.01	0.02	0.94
	$T=30$ s	-0.26	-0.22	-0.05	-1.07	-1.04	-0.22	0.48	0.39	0.69	0.01	0.01	0.98
	$T=60$ s	-0.02	-0.04	-0.14	-1.01	-0.98	-0.51	0.15	0.17	0.94	0.02	0.00	1.00
7 km	$T=15$ s	-0.30	-0.10	-0.19	-1.41	-1.42	-0.23	0.55	0.46	0.86	0.01	0.02	0.97
	$T=30$ s	-0.40	-0.17	-0.11	-1.52	-1.46	-0.30	0.46	0.38	0.84	0.01	0.01	0.98
	$T=60$ s	-0.08	-0.13	-0.34	-1.37	-1.28	-0.64	0.13	0.14	0.99	0.01	0.01	1.00
CWNA/C2													
3 km	$T=15$ s	0.00	-0.01	-0.19	-0.07	-0.12	-0.17	0.04	0.09	0.96	0.01	0.00	0.89
	$T=30$ s	0.01	-0.02	-0.03	-0.28	-0.32	-0.13	0.07	0.07	0.53	0.01	0.01	0.92
	$T=60$ s	0.02	0.00	0.02	-0.58	-0.57	-0.34	0.06	0.03	0.84	0.01	0.01	1.00
5 km	$T=15$ s	0.04	-0.09	-0.31	0.28	0.22	-0.18	0.25	0.37	0.99	0.02	0.02	0.62
	$T=30$ s	0.06	-0.07	-0.10	-0.20	-0.25	-0.18	0.05	0.11	0.80	0.01	0.01	0.96
	$T=60$ s	0.08	0.01	-0.12	-0.84	-0.82	-0.47	0.03	0.02	0.98	0.01	0.00	1.00
7 km	$T=15$ s	0.04	-0.03	-0.40	0.70	0.59	-0.19	0.46	0.60	1.00	0.04	0.03	0.46
	$T=30$ s	-0.00	-0.03	-0.17	0.05	0.00	-0.22	0.12	0.21	0.91	0.01	0.01	0.96
	$T=60$ s	0.07	-0.10	-0.30	-1.08	-1.04	-0.58	0.01	0.02	0.99	0.01	0.01	1.00

Table C.2: Consistency summary statistics for the CWNA model applied to targets moving in rolling terrain (T2).

		$\mathbf{err}_{\mu_x}$ (m)	$\mathbf{err}_{\mu_y}$ (m)	$\mathbf{err}_{\mu_s}$ (m/s)	$\mathbf{err}_{\sigma_x}$ (m)	$\mathbf{err}_{\sigma_y}$ (m)	$\mathbf{err}_{\sigma_s}$ (m/s)	$\mathbf{rej}_{\mu_x}$ (%)	$\mathbf{rej}_{\mu_y}$ (%)	$\mathbf{rej}_{\mu_s}$ (%)	$\mathbf{rej}_{\sigma_x}$ (%)	$\mathbf{rej}_{\sigma_y}$ (%)	$\mathbf{rej}_{\sigma_s}$ (%)
CWNA/C1													
3 km	$T=15$ s	0.02	-0.04	-0.18	-0.68	-0.71	-0.20	0.38	0.45	0.91	0.02	0.01	0.93
	$T=30$ s	-0.00	-0.03	-0.09	-0.69	-0.68	-0.16	0.39	0.49	0.55	0.02	0.02	0.93
	$T=60$ s	-0.01	-0.03	-0.14	-0.64	-0.68	-0.41	0.19	0.22	0.92	0.02	0.02	0.99
5 km	$T=15$ s	0.08	-0.04	-0.23	-1.09	-1.11	-0.22	0.43	0.53	0.88	0.02	0.02	0.94
	$T=30$ s	-0.00	0.01	-0.13	-1.11	-1.11	-0.24	0.36	0.48	0.79	0.02	0.02	0.98
	$T=60$ s	0.04	-0.09	-0.29	-0.96	-1.02	-0.54	0.13	0.18	0.96	0.01	0.02	1.00
7 km	$T=15$ s	0.08	-0.07	-0.26	-1.49	-1.50	-0.25	0.46	0.55	0.89	0.02	0.02	0.97
	$T=30$ s	0.07	-0.09	-0.19	-1.58	-1.61	-0.32	0.36	0.46	0.86	0.03	0.03	0.99
	$T=60$ s	-0.03	-0.12	-0.50	-1.30	-1.34	-0.67	0.08	0.14	0.99	0.02	0.02	1.00
CWNA/C2													
3 km	$T=15$ s	-0.02	-0.01	-0.25	-0.24	-0.21	-0.21	0.07	0.06	0.97	0.01	0.01	0.96
	$T=30$ s	-0.04	-0.01	-0.10	-0.45	-0.40	-0.17	0.06	0.06	0.55	0.01	0.01	0.98
	$T=60$ s	-0.04	0.02	-0.11	-0.59	-0.61	-0.38	0.04	0.04	0.89	0.02	0.02	0.99
5 km	$T=15$ s	0.01	-0.02	-0.37	-0.02	0.03	-0.23	0.29	0.23	0.99	0.01	0.01	0.81
	$T=30$ s	-0.06	0.03	-0.17	-0.42	-0.39	-0.22	0.09	0.06	0.87	0.01	0.02	0.98
	$T=60$ s	-0.02	-0.03	-0.25	-0.83	-0.86	-0.50	0.04	0.02	0.95	0.01	0.03	1.00
7 km	$T=15$ s	-0.03	-0.04	-0.46	0.26	0.31	-0.24	0.55	0.37	0.99	0.02	0.02	0.62
	$T=30$ s	-0.01	-0.08	-0.24	-0.33	-0.31	-0.26	0.20	0.13	0.90	0.02	0.02	0.98
	$T=60$ s	-0.09	-0.08	-0.44	-1.08	-1.09	-0.61	0.02	0.01	0.98	0.01	0.02	1.00

Table C.3: Consistency summary statistics for the CWNA model applied to targets moving in mountain terrain (T3).

		$\mathbf{error_{xy}}$ (m)		$\mathbf{error_{s}}$ (m/s)		**divergence**		$\triangle\mathbf{error_{xy}}$ (%)		**area** (m^2)		$\triangle$**area** (%)	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
CWNA/C1													
3 km	$T=15$ s	7.66	0.24	0.176	0.024	5.30	0.30	-9.27	3.39	168.38	2.00	-25.93	1.21
	$T=30$ s	8.20	0.17	0.127	0.010	2.55	0.16	-3.38	3.15	191.68	1.20	-15.99	0.39
	$T=60$ s	8.46	0.04	0.107	0.004	1.28	0.02	-1.48	0.39	217.98	0.74	-4.46	0.14
5 km	$T=15$ s	11.19	0.64	0.249	0.078	5.78	0.61	-13.51	5.35	259.18	3.02	-31.34	0.97
	$T=30$ s	12.34	0.41	0.159	0.023	2.67	0.11	-6.22	3.53	316.15	3.22	-16.36	0.99
	$T=60$ s	13.08	0.09	0.180	0.010	1.39	0.01	-2.40	0.38	360.06	1.14	-4.84	0.16
7 km	$T=15$ s	14.41	0.81	0.300	0.106	5.83	0.48	-17.99	4.79	353.01	2.14	-33.18	0.47
	$T=30$ s	16.39	0.58	0.213	0.045	2.80	0.11	-8.57	3.64	438.68	5.20	-17.00	1.06
	$T=60$ s	17.88	0.14	0.347	0.028	1.46	0.01	-2.41	0.73	504.33	1.93	-4.63	0.37
CWNA/C2													
3 km	$T=15$ s	7.67	0.10	0.194	0.015	5.10	0.05	-11.30	1.23	168.50	2.48	-25.83	1.56
	$T=30$ s	7.94	0.06	0.120	0.005	2.41	0.02	-8.23	0.20	190.02	0.51	-16.67	0.19
	$T=60$ s	8.41	0.06	0.103	0.004	1.27	0.02	-2.53	0.16	216.95	0.92	-4.88	0.05
5 km	$T=15$ s	11.37	0.11	0.322	0.064	5.90	0.08	-14.85	1.24	254.11	6.05	-32.58	1.86
	$T=30$ s	12.22	0.12	0.172	0.010	2.65	0.01	-9.23	0.95	309.18	4.89	-18.13	1.54
	$T=60$ s	13.06	0.05	0.173	0.006	1.38	0.00	-3.08	0.28	358.32	1.30	-5.27	0.24
7 km	$T=15$ s	14.96	0.35	0.430	0.095	6.51	0.08	-17.44	1.99	327.64	10.72	-37.86	2.23
	$T=30$ s	16.51	0.16	0.252	0.034	2.85	0.03	-9.97	1.20	420.39	8.93	-20.39	1.85
	$T=60$ s	17.93	0.20	0.328	0.026	1.46	0.01	-2.62	0.88	501.85	2.61	-5.08	0.54
CWNA/C3													
3 km	$T=15$ s	9.01	0.23	2.380	0.061	6.58	0.12	5.10	3.47	225.17	2.91	-1.29	1.45
	$T=30$ s	8.85	0.08	2.103	0.010	3.15	0.01	2.08	0.98	227.19	1.21	-0.43	0.58
	$T=60$ s	8.59	0.04	3.794	0.003	1.67	0.02	-0.43	0.13	224.50	1.06	-1.61	0.09
5 km	$T=15$ s	13.53	0.36	2.642	0.118	6.93	0.15	1.93	3.59	359.91	12.47	-4.91	3.59
	$T=30$ s	14.05	0.51	2.234	0.056	3.36	0.06	4.98	4.61	372.74	2.45	-1.40	0.70
	$T=60$ s	13.62	0.14	3.858	0.009	1.74	0.01	1.09	0.98	374.56	0.92	-1.03	0.12
7 km	$T=15$ s	17.86	0.37	2.869	0.155	7.14	0.17	-1.41	2.70	490.29	22.92	-7.34	4.61
	$T=30$ s	18.91	0.61	2.358	0.083	3.48	0.09	3.57	4.23	511.99	3.20	-3.18	0.58
	$T=60$ s	19.04	0.51	4.023	0.173	1.80	0.02	3.81	3.29	526.44	3.83	-0.45	0.80

Table C.4: State estimation performance summary statistics for the CWNA model applied to targets moving in flat terrain (T1).

		$\mathbf{error_{xy}}$ (m)		$\mathbf{error_{s}}$ (m/s)		**divergence**		$\triangle\mathbf{error_{xy}}$ (%)		**area** (m^2)		$\triangle$**area** (%)	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
CWNA/C1													
3 km	$T=15$ s	8.01	0.13	0.217	0.016	5.57	0.10	-4.49	1.35	172.85	1.88	-23.87	0.95
	$T=30$ s	8.56	0.06	0.176	0.003	2.76	0.02	1.17	0.64	194.80	0.72	-14.32	0.42
	$T=60$ s	8.47	0.04	0.138	0.004	1.35	0.02	-1.19	0.17	217.71	0.29	-4.19	0.05
5 km	$T=15$ s	11.70	0.20	0.264	0.030	5.98	0.10	-9.26	1.40	264.13	2.89	-30.01	0.82
	$T=30$ s	12.65	0.15	0.203	0.010	2.78	0.03	-3.16	0.99	317.84	1.39	-15.83	0.42
	$T=60$ s	13.13	0.12	0.213	0.008	1.39	0.00	-1.73	0.69	360.83	0.70	-4.43	0.23
7 km	$T=15$ s	14.89	0.20	0.293	0.038	6.01	0.09	-14.90	1.23	354.03	2.52	-32.97	0.50
	$T=30$ s	16.55	0.28	0.232	0.018	2.84	0.03	-7.01	1.46	437.33	2.23	-17.20	0.44
	$T=60$ s	17.93	0.20	0.384	0.011	1.47	0.01	-1.79	1.20	504.14	1.58	-4.54	0.34
CWNA/C2													
3 km	$T=15$ s	7.73	0.05	0.239	0.015	5.27	0.06	-10.03	0.78	173.99	1.61	-23.35	0.83
	$T=30$ s	8.03	0.04	0.157	0.002	2.58	0.01	-6.81	0.33	192.61	0.18	-15.25	0.18
	$T=60$ s	8.42	0.03	0.136	0.003	1.35	0.02	-2.22	0.12	217.28	0.45	-4.38	0.05
5 km	$T=15$ s	11.43	0.12	0.343	0.023	5.94	0.06	-14.30	0.92	261.27	3.60	-30.73	1.02
	$T=30$ s	12.19	0.07	0.201	0.008	2.71	0.01	-9.02	0.54	309.13	2.51	-18.11	0.72
	$T=60$ s	13.09	0.09	0.196	0.005	1.39	0.00	-2.57	0.56	359.19	0.35	-4.86	0.15
7 km	$T=15$ s	14.97	0.24	0.430	0.030	6.45	0.05	-17.27	1.63	335.09	6.56	-36.51	1.29
	$T=30$ s	16.36	0.17	0.256	0.014	2.87	0.02	-10.46	1.01	416.36	4.12	-21.13	0.82
	$T=60$ s	17.93	0.13	0.343	0.009	1.46	0.00	-2.38	0.79	501.13	1.26	-5.10	0.29

Table C.5: State estimation performance summary statistics for the CWNA model applied to targets moving in rolling terrain (T2).

		$\mathbf{error_{xy}}$ (m)		$\mathbf{error_{s}}$ (m/s)		**divergence**		$\mathbf{\triangle error_{xy}}$ (%)		**area** (m^2)		$\mathbf{\triangle area}$ (%)	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
CWNA/C1													
3 km	$T=15$ s	7.86	0.15	0.262	0.029	5.52	0.11	-6.72	1.20	173.48	4.01	-23.71	0.98
	$T=30$ s	8.32	0.08	0.233	0.006	2.80	0.04	-1.79	1.07	191.42	2.69	-15.87	0.48
	$T=60$ s	8.36	0.08	0.237	0.006	1.33	0.01	-2.82	0.60	214.61	1.86	-5.88	0.25
5 km	$T=15$ s	11.45	0.22	0.314	0.059	5.95	0.20	-11.39	1.68	264.25	3.80	-29.97	0.72
	$T=30$ s	12.39	0.22	0.258	0.009	2.84	0.02	-5.12	1.65	310.69	4.45	-17.73	0.87
	$T=60$ s	12.92	0.09	0.331	0.011	1.41	0.01	-3.42	0.29	353.38	1.51	-6.47	0.20
7 km	$T=15$ s	14.67	0.25	0.342	0.071	6.09	0.19	-16.22	1.60	349.35	3.50	-33.84	0.50
	$T=30$ s	16.24	0.30	0.293	0.025	2.90	0.04	-8.54	1.67	426.08	4.49	-19.32	0.69
	$T=60$ s	17.66	0.27	0.515	0.012	1.50	0.01	-3.27	1.31	492.94	2.99	-6.69	0.47
CWNA/C2													
3 km	$T=15$ s	7.70	0.13	0.287	0.021	5.27	0.08	-10.66	0.94	176.42	4.46	-22.44	1.14
	$T=30$ s	7.94	0.02	0.208	0.001	2.58	0.01	-7.94	0.36	193.78	2.02	-14.93	0.19
	$T=60$ s	8.33	0.07	0.224	0.003	1.33	0.01	-3.56	0.45	214.35	1.84	-6.01	0.28
5 km	$T=15$ s	11.27	0.22	0.398	0.039	5.86	0.12	-15.40	1.62	263.56	6.94	-30.14	1.55
	$T=30$ s	12.07	0.10	0.252	0.010	2.73	0.01	-9.83	0.70	309.03	4.29	-18.18	0.85
	$T=60$ s	12.92	0.07	0.299	0.006	1.41	0.01	-3.91	0.19	352.76	1.18	-6.65	0.11
7 km	$T=15$ s	14.58	0.43	0.485	0.043	6.28	0.12	-19.32	2.48	335.67	11.56	-36.40	2.05
	$T=30$ s	16.05	0.23	0.314	0.022	2.88	0.03	-11.87	1.29	413.16	5.38	-21.77	0.87
	$T=60$ s	17.69	0.19	0.454	0.007	1.48	0.01	-3.65	0.86	491.86	2.63	-6.90	0.38

Table C.6: State estimation performance summary statistics for the CWNA model applied to targets moving in mountain terrain (T3).

		$\mathbf{err}_{\mu_x}$ (m)	$\mathbf{err}_{\mu_y}$ (m)	$\mathbf{err}_{\mu_s}$ (m/s)	$\mathbf{err}_{\sigma_x}$ (m)	$\mathbf{err}_{\sigma_y}$ (m)	$\mathbf{err}_{\sigma_s}$ (m/s)	$\mathbf{rej}_{\mu_x}$ (%)	$\mathbf{rej}_{\mu_y}$ (%)	$\mathbf{rej}_{\mu_s}$ (%)	$\mathbf{rej}_{\sigma_x}$ (%)	$\mathbf{rej}_{\sigma_y}$ (%)	$\mathbf{rej}_{\sigma_s}$ (%)
PSH/C1													
3 km	$T=15$ s	0.02	-0.19	0.11	-0.52	-0.51	-0.06	0.41	0.39	0.53	0.02	0.02	0.55
	$T=30$ s	0.05	-0.30	0.09	-0.42	-0.36	-0.14	0.29	0.32	0.45	0.05	0.03	0.80
	$T=60$ s	0.40	-0.06	-0.16	-0.14	-0.11	-1.17	0.10	0.17	0.92	0.03	0.04	0.92
5 km	$T=15$ s	0.11	-0.31	0.16	-0.91	-0.87	-0.08	0.54	0.70	0.92	0.02	0.02	0.57
	$T=30$ s	-0.02	-0.34	0.07	-0.65	-0.55	-0.20	0.34	0.34	0.60	0.03	0.04	0.72
	$T=60$ s	0.66	-0.17	-0.17	-0.50	-0.34	-1.17	0.09	0.10	0.89	0.03	0.03	0.97
7 km	$T=15$ s	0.11	-0.56	0.17	-1.21	-1.07	-0.11	0.54	0.71	0.92	0.02	0.02	0.56
	$T=30$ s	0.12	-0.46	0.06	-0.95	-0.86	-0.27	0.38	0.47	0.60	0.03	0.03	0.72
	$T=60$ s	0.50	-0.25	-0.17	-0.78	-0.57	-1.15	0.12	0.19	0.92	0.03	0.02	0.98
PSH/C2													
3 km	$T=15$ s	0.04	-0.04	-0.01	-0.12	-0.23	-0.05	0.31	0.30	0.18	0.00	0.01	0.44
	$T=30$ s	0.04	-0.06	-0.02	-0.35	-0.42	-0.15	0.04	0.02	0.74	0.01	0.01	0.90
	$T=60$ s	0.03	0.03	-0.30	-0.57	-0.60	-0.98	0.03	0.01	0.99	0.01	0.01	0.99
5 km	$T=15$ s	0.11	0.01	0.02	-0.19	-0.32	-0.05	0.49	0.67	0.12	0.00	0.00	0.35
	$T=30$ s	0.11	-0.08	-0.07	-0.47	-0.53	-0.20	0.12	0.15	0.70	0.01	0.01	0.79
	$T=60$ s	0.20	0.02	-0.31	-0.88	-0.85	-0.98	0.01	0.01	0.97	0.01	0.00	0.98
7 km	$T=15$ s	0.09	-0.08	0.01	-0.25	-0.38	-0.08	0.49	0.68	0.09	0.01	0.01	0.43
	$T=30$ s	0.13	-0.13	-0.11	-0.67	-0.75	-0.26	0.31	0.43	0.68	0.00	0.00	0.79
	$T=60$ s	0.14	-0.07	-0.32	-1.14	-1.18	-0.99	0.00	0.01	0.97	0.02	0.01	0.99
PSH/C3													
3 km	$T=15$ s	14.47	-19.39	0.93	13.74	12.00	0.10	0.83	0.96	0.98	0.97	0.98	0.72
	$T=30$ s	14.67	-18.81	0.68	4.16	4.20	0.10	0.93	0.78	0.99	0.69	0.67	0.70
	$T=60$ s	8.70	-18.32	0.30	1.14	0.87	-0.06	0.88	0.69	0.99	0.19	0.22	0.38
5 km	$T=15$ s	21.45	-23.39	0.93	23.74	25.07	0.11	0.81	0.96	1.00	0.96	0.97	0.69
	$T=30$ s	20.61	-28.18	0.69	13.76	9.43	0.10	0.83	0.90	0.97	0.84	0.82	0.64
	$T=60$ s	16.18	-32.32	0.39	4.23	2.75	-0.08	0.85	0.73	0.99	0.39	0.37	0.41
7 km	$T=15$ s	20.47	-17.95	0.96	35.87	38.54	0.12	0.80	0.93	1.00	0.97	0.96	0.66
	$T=30$ s	23.32	-35.35	0.70	24.86	20.72	0.13	0.85	0.90	0.98	0.90	0.91	0.69
	$T=60$ s	17.63	-44.06	0.40	8.80	5.72	-0.10	0.85	0.73	0.97	0.61	0.52	0.45

Table C.7: Consistency summary statistics for the PSH model applied to targets moving in flat terrain (T1).

		$\mathbf{err}_{\mu_x}$ (m)	$\mathbf{err}_{\mu_y}$ (m)	$\mathbf{err}_{\mu_s}$ (m/s)	$\mathbf{err}_{\sigma_x}$ (m)	$\mathbf{err}_{\sigma_y}$ (m)	$\mathbf{err}_{\sigma_s}$ (m/s)	$\mathbf{rej}_{\mu_x}$ (%)	$\mathbf{rej}_{\mu_y}$ (%)	$\mathbf{rej}_{\mu_s}$ (%)	$\mathbf{rej}_{\sigma_x}$ (%)	$\mathbf{rej}_{\sigma_y}$ (%)	$\mathbf{rej}_{\sigma_s}$ (%)
PSH/C4													
3 km	$T=15$ s	0.06	-0.02	0.04	0.75	0.49	-0.14	0.40	0.60	0.74	0.31	0.23	0.64
	$T=30$ s	0.09	0.05	0.01	0.37	0.19	-0.21	0.18	0.18	0.93	0.13	0.14	0.52
	$T=60$ s	0.01	0.11	-0.01	-0.15	-0.26	-0.24	0.21	0.17	1.00	0.03	0.04	0.13
5 km	$T=15$ s	0.14	0.01	0.08	1.27	1.00	-0.09	0.54	0.70	0.84	0.39	0.28	0.66
	$T=30$ s	0.19	0.02	0.03	1.22	0.90	-0.17	0.28	0.48	0.82	0.30	0.24	0.45
	$T=60$ s	0.14	0.16	-0.01	0.30	0.17	-0.24	0.22	0.19	0.97	0.10	0.11	0.24
7 km	$T=15$ s	0.09	-0.07	0.10	1.67	1.34	-0.07	0.58	0.72	0.89	0.45	0.35	0.72
	$T=30$ s	0.16	-0.13	0.05	2.05	1.74	-0.12	0.42	0.63	0.74	0.39	0.33	0.47
	$T=60$ s	0.14	0.04	-0.00	1.10	0.75	-0.24	0.14	0.22	0.92	0.20	0.21	0.22

Continuation of Table C.7: Consistency summary statistics for the PSH model applied to targets moving in flat terrain (T1).

		$\mathbf{err}_{\mu_x}$ (m)	$\mathbf{err}_{\mu_y}$ (m)	$\mathbf{err}_{\mu_s}$ (m/s)	$\mathbf{err}_{\sigma_x}$ (m)	$\mathbf{err}_{\sigma_y}$ (m)	$\mathbf{err}_{\sigma_s}$ (m/s)	$\mathbf{rej}_{\mu_x}$ (%)	$\mathbf{rej}_{\mu_y}$ (%)	$\mathbf{rej}_{\mu_s}$ (%)	$\mathbf{rej}_{\sigma_x}$ (%)	$\mathbf{rej}_{\sigma_y}$ (%)	$\mathbf{rej}_{\sigma_s}$ (%)
PSH/C1													
3 km	$T=15$ s	-0.12	-0.02	0.18	-0.29	-0.31	-0.04	0.61	0.61	0.68	0.02	0.03	0.39
	$T=30$ s	-0.01	-0.27	0.22	0.17	0.26	-0.09	0.61	0.58	0.73	0.09	0.09	0.55
	$T=60$ s	-0.19	0.18	0.13	0.20	0.31	-0.82	0.24	0.34	0.94	0.06	0.07	0.97
5 km	$T=15$ s	-0.07	0.08	0.22	-0.56	-0.58	-0.06	0.70	0.76	0.86	0.02	0.03	0.38
	$T=30$ s	-0.06	0.05	0.22	0.00	0.19	-0.15	0.54	0.64	0.73	0.06	0.07	0.55
	$T=60$ s	-0.18	0.30	0.09	-0.05	0.12	-0.86	0.25	0.39	0.93	0.05	0.05	0.97
7 km	$T=15$ s	-0.08	0.18	0.22	-0.79	-0.78	-0.09	0.73	0.82	0.87	0.04	0.03	0.37
	$T=30$ s	-0.20	0.21	0.20	-0.21	0.10	-0.20	0.53	0.67	0.76	0.05	0.05	0.56
	$T=60$ s	-0.17	0.24	0.09	-0.18	-0.02	-0.87	0.25	0.48	0.89	0.04	0.04	0.99
PSH/C2													
3 km	$T=15$ s	0.00	0.01	0.01	-0.23	-0.29	-0.05	0.28	0.45	0.20	0.00	0.00	0.40
	$T=30$ s	-0.01	-0.03	0.02	-0.34	-0.39	-0.15	0.09	0.10	0.58	0.00	0.01	0.82
	$T=60$ s	0.00	-0.02	-0.22	-0.60	-0.58	-0.90	0.03	0.03	0.99	0.01	0.01	0.99
5 km	$T=15$ s	0.04	-0.06	0.02	-0.37	-0.46	-0.06	0.60	0.72	0.17	0.00	0.01	0.35
	$T=30$ s	0.02	-0.09	-0.03	-0.48	-0.54	-0.18	0.10	0.21	0.52	0.00	0.01	0.74
	$T=60$ s	0.05	-0.01	-0.24	-0.89	-0.85	-0.91	0.02	0.02	0.98	0.02	0.00	0.99
7 km	$T=15$ s	0.05	0.01	0.01	-0.50	-0.62	-0.09	0.65	0.81	0.16	0.00	0.01	0.39
	$T=30$ s	-0.00	-0.06	-0.07	-0.67	-0.71	-0.23	0.27	0.42	0.54	0.01	0.00	0.74
	$T=60$ s	0.03	-0.09	-0.24	-1.19	-1.13	-0.91	0.04	0.06	0.98	0.00	0.01	0.98
PSH/C3													
3 km	$T=15$ s	-14.35	-1.94	0.86	15.17	18.19	0.17	0.80	0.95	1.00	0.94	0.99	0.77
	$T=30$ s	-17.61	-10.15	0.61	6.57	8.79	0.09	0.75	0.95	0.99	0.74	0.87	0.63
	$T=60$ s	-16.46	-5.97	0.27	1.14	2.36	-0.10	0.76	0.93	0.89	0.26	0.44	0.46
5 km	$T=15$ s	-15.54	-4.43	0.84	29.78	27.69	0.16	0.78	0.94	0.99	0.95	0.99	0.73
	$T=30$ s	-12.91	-11.52	0.66	20.23	17.66	0.12	0.74	0.94	0.99	0.88	0.96	0.67
	$T=60$ s	-25.62	-9.14	0.38	6.87	8.74	-0.08	0.75	0.95	0.90	0.62	0.70	0.44
7 km	$T=15$ s	-10.47	-10.01	0.82	41.81	37.15	0.16	0.81	0.91	0.99	0.95	0.99	0.70
	$T=30$ s	-32.74	-12.93	0.64	24.91	25.20	0.10	0.77	0.90	0.98	0.90	0.96	0.67
	$T=60$ s	-33.34	-10.04	0.42	14.92	16.41	-0.09	0.80	0.92	0.96	0.80	0.83	0.52

Table C.8: Consistency summary statistics for the PSH model applied to targets moving in rolling terrain (T2).

		$\mathbf{err}_{\mu_x}$ (m)	$\mathbf{err}_{\mu_y}$ (m)	$\mathbf{err}_{\mu_s}$ (m/s)	$\mathbf{err}_{\sigma_x}$ (m)	$\mathbf{err}_{\sigma_y}$ (m)	$\mathbf{err}_{\sigma_s}$ (m/s)	$\mathbf{rej}_{\mu_x}$ (%)	$\mathbf{rej}_{\mu_y}$ (%)	$\mathbf{rej}_{\mu_s}$ (%)	$\mathbf{rej}_{\sigma_x}$ (%)	$\mathbf{rej}_{\sigma_y}$ (%)	$\mathbf{rej}_{\sigma_s}$ (%)
PSH/C4													
3 km	$T=15$ s	-0.03	0.06	0.05	0.74	0.54	-0.13	0.46	0.54	0.72	0.30	0.21	0.55
	$T=30$ s	-0.04	0.00	0.03	0.53	0.36	-0.21	0.25	0.22	0.94	0.19	0.14	0.60
	$T=60$ s	-0.20	0.33	0.00	0.09	-0.03	-0.26	0.21	0.30	0.98	0.05	0.05	0.30
5 km	$T=15$ s	0.01	-0.02	0.09	1.26	1.01	-0.09	0.67	0.78	0.81	0.36	0.28	0.61
	$T=30$ s	-0.06	-0.03	0.05	1.44	1.23	-0.16	0.30	0.39	0.76	0.34	0.27	0.52
	$T=60$ s	-0.25	0.36	0.01	0.80	0.65	-0.25	0.17	0.18	0.95	0.19	0.14	0.35
7 km	$T=15$ s	0.02	0.05	0.11	1.62	1.32	-0.08	0.72	0.87	0.85	0.39	0.31	0.63
	$T=30$ s	-0.09	-0.01	0.07	2.31	2.09	-0.12	0.50	0.65	0.70	0.47	0.39	0.57
	$T=60$ s	-0.23	0.31	0.02	1.74	1.55	-0.24	0.15	0.06	0.90	0.31	0.26	0.32

Continuation of Table C.8: Consistency summary statistics for the PSH model applied to targets moving in rolling terrain (T2).

		$\mathbf{err}_{\mu_x}$ (m)	$\mathbf{err}_{\mu_y}$ (m)	$\mathbf{err}_{\mu_s}$ (m/s)	$\mathbf{err}_{\sigma_x}$ (m)	$\mathbf{err}_{\sigma_y}$ (m)	$\mathbf{err}_{\sigma_s}$ (m/s)	$\mathbf{rej}_{\mu_x}$ (%)	$\mathbf{rej}_{\mu_y}$ (%)	$\mathbf{rej}_{\mu_s}$ (%)	$\mathbf{rej}_{\sigma_x}$ (%)	$\mathbf{rej}_{\sigma_y}$ (%)	$\mathbf{rej}_{\sigma_s}$ (%)
PSH/C1													
3 km	$T=15$ s	0.03	-0.03	0.17	-0.37	-0.40	-0.07	0.65	0.60	0.73	0.03	0.03	0.32
	$T=30$ s	0.24	-0.38	0.22	0.36	0.26	-0.10	0.50	0.56	0.70	0.13	0.12	0.46
	$T=60$ s	0.20	-0.12	0.22	0.67	0.54	-0.59	0.39	0.35	0.87	0.11	0.10	0.94
5 km	$T=15$ s	0.11	-0.01	0.20	-0.59	-0.72	-0.09	0.76	0.68	0.91	0.02	0.02	0.34
	$T=30$ s	0.17	-0.29	0.21	0.44	0.18	-0.14	0.61	0.55	0.71	0.11	0.10	0.45
	$T=60$ s	0.19	-0.29	0.17	0.47	0.28	-0.64	0.35	0.39	0.85	0.09	0.08	0.93
7 km	$T=15$ s	0.11	0.01	0.21	-0.72	-0.96	-0.11	0.80	0.67	0.91	0.03	0.03	0.37
	$T=30$ s	0.31	-0.41	0.20	0.35	-0.13	-0.19	0.69	0.57	0.67	0.10	0.09	0.51
	$T=60$ s	0.26	-0.50	0.14	0.17	-0.08	-0.68	0.40	0.34	0.81	0.05	0.07	0.93
PSH/C2													
3 km	$T=15$ s	-0.01	-0.03	-0.02	-0.33	-0.31	-0.07	0.39	0.29	0.21	0.00	0.00	0.32
	$T=30$ s	-0.04	-0.01	0.00	-0.45	-0.41	-0.16	0.06	0.06	0.47	0.01	0.01	0.75
	$T=60$ s	-0.02	0.01	-0.21	-0.59	-0.62	-0.85	0.03	0.03	0.98	0.02	0.01	0.98
5 km	$T=15$ s	0.02	-0.04	-0.01	-0.52	-0.46	-0.08	0.66	0.52	0.21	0.01	0.01	0.26
	$T=30$ s	-0.06	0.04	-0.03	-0.59	-0.58	-0.18	0.17	0.12	0.48	0.01	0.02	0.67
	$T=60$ s	0.00	-0.05	-0.22	-0.86	-0.90	-0.84	0.04	0.03	0.97	0.01	0.02	0.96
7 km	$T=15$ s	-0.02	-0.06	-0.02	-0.69	-0.62	-0.11	0.72	0.58	0.21	0.01	0.01	0.34
	$T=30$ s	-0.03	-0.06	-0.07	-0.84	-0.82	-0.22	0.40	0.28	0.53	0.01	0.01	0.67
	$T=60$ s	-0.09	-0.09	-0.22	-1.15	-1.19	-0.83	0.03	0.02	0.93	0.01	0.02	0.96
PSH/C3													
3 km	$T=15$ s	23.57	6.23	0.77	6.59	6.38	0.05	0.92	0.96	0.99	0.78	0.89	0.65
	$T=30$ s	18.10	-2.74	0.49	2.46	2.96	-0.03	0.75	0.93	0.95	0.50	0.65	0.65
	$T=60$ s	9.84	-8.20	0.24	0.12	-0.19	-0.20	0.86	0.91	0.96	0.12	0.09	0.46
5 km	$T=15$ s	25.85	6.20	0.75	20.38	17.01	0.04	0.92	0.94	0.98	0.90	0.97	0.65
	$T=30$ s	25.06	-3.61	0.53	10.66	9.47	-0.02	0.79	0.93	0.96	0.74	0.83	0.67
	$T=60$ s	18.87	-15.51	0.30	1.69	1.02	-0.21	0.86	0.87	0.97	0.29	0.27	0.49
7 km	$T=15$ s	25.61	5.64	0.74	33.62	29.10	0.05	0.91	0.93	0.98	0.93	0.98	0.59
	$T=30$ s	24.52	-2.56	0.54	22.50	17.58	-0.02	0.85	0.91	0.94	0.81	0.91	0.65
	$T=60$ s	24.22	-21.05	0.31	5.98	4.72	-0.21	0.80	0.85	0.97	0.37	0.37	0.48

Table C.9: Consistency summary statistics for the PSH model applied to targets moving in mountain terrain (T3).

		$\mathbf{err}_{\mu_x}$ (m)	$\mathbf{err}_{\mu_y}$ (m)	$\mathbf{err}_{\mu_s}$ (m/s)	$\mathbf{err}_{\sigma_x}$ (m)	$\mathbf{err}_{\sigma_y}$ (m)	$\mathbf{err}_{\sigma_s}$ (m/s)	$\mathbf{rej}_{\mu_x}$ (%)	$\mathbf{rej}_{\mu_y}$ (%)	$\mathbf{rej}_{\mu_s}$ (%)	$\mathbf{rej}_{\sigma_x}$ (%)	$\mathbf{rej}_{\sigma_y}$ (%)	$\mathbf{rej}_{\sigma_s}$ (%)
PSH/C4													
3 km	$T=15$ s	-0.07	-0.01	0.05	0.40	0.57	-0.18	0.55	0.39	0.79	0.16	0.24	0.55
	$T=30$ s	-0.08	-0.00	0.04	0.19	0.37	-0.25	0.19	0.12	0.88	0.11	0.12	0.60
	$T=60$ s	-0.14	0.09	0.03	-0.07	-0.00	-0.30	0.23	0.18	0.99	0.05	0.04	0.36
5 km	$T=15$ s	-0.04	-0.03	0.08	0.79	0.99	-0.14	0.73	0.59	0.86	0.23	0.32	0.55
	$T=30$ s	-0.10	0.04	0.06	1.00	1.22	-0.20	0.44	0.27	0.78	0.25	0.29	0.54
	$T=60$ s	-0.13	0.02	0.03	0.48	0.62	-0.29	0.14	0.12	0.92	0.11	0.15	0.34
7 km	$T=15$ s	-0.07	-0.04	0.10	1.06	1.29	-0.14	0.75	0.63	0.90	0.25	0.32	0.58
	$T=30$ s	-0.07	-0.11	0.08	1.69	1.91	-0.17	0.63	0.48	0.80	0.31	0.37	0.56
	$T=60$ s	-0.17	-0.05	0.04	1.29	1.54	-0.27	0.13	0.07	0.81	0.22	0.27	0.31

Continuation of Table C.9: Consistency summary statistics for the PSH model applied to targets moving in mountain terrain (T3).

		$\mathbf{error_{xy}}$ (m)		$\mathbf{error_{s}}$ (m/s)		**divergence**		$\mathbf{\triangle error_{xy}}$ (%)		**area** (m^2)		$\mathbf{\triangle area}$ (%)	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
PSH/C1													
3 km	$T=15$ s	7.82	0.20	0.217	0.027	6.11	0.31	-7.04	2.98	173.15	3.92	-23.77	2.25
	$T=30$ s	8.58	0.13	0.185	0.009	3.02	0.04	0.66	1.45	196.34	0.65	-13.91	0.34
	$T=60$ s	9.08	0.19	0.404	0.007	1.59	0.06	5.81	2.19	220.13	0.43	-3.50	0.28
5 km	$T=15$ s	11.86	0.77	0.281	0.067	6.66	0.70	-7.55	6.97	268.57	2.37	-28.84	0.82
	$T=30$ s	13.09	0.33	0.234	0.032	3.13	0.14	-0.50	2.77	323.57	4.37	-14.35	1.32
	$T=60$ s	13.87	0.23	0.400	0.018	1.57	0.03	3.83	1.92	361.79	1.10	-4.36	0.33
7 km	$T=15$ s	16.01	1.63	0.325	0.087	7.05	1.17	-7.87	10.38	357.58	4.52	-32.32	0.89
	$T=30$ s	17.40	0.53	0.287	0.044	3.21	0.16	-2.54	3.45	443.40	5.75	-16.09	1.13
	$T=60$ s	18.57	0.38	0.405	0.020	1.59	0.05	1.83	2.06	501.41	2.03	-5.18	0.46
PSH/C2													
3 km	$T=15$ s	7.79	0.15	0.203	0.022	5.47	0.21	-9.30	2.32	174.56	4.44	-23.14	2.50
	$T=30$ s	8.08	0.06	0.129	0.006	2.47	0.02	-6.60	0.27	198.86	0.69	-12.86	0.18
	$T=60$ s	8.52	0.05	0.325	0.003	1.39	0.02	-1.28	0.08	221.54	1.07	-2.85	0.08
5 km	$T=15$ s	11.78	0.46	0.288	0.051	6.11	0.33	-10.21	4.59	266.37	5.04	-29.38	1.53
	$T=30$ s	12.60	0.22	0.213	0.022	2.77	0.06	-6.14	1.88	329.34	5.51	-12.83	1.65
	$T=60$ s	13.19	0.05	0.340	0.005	1.46	0.00	-2.04	0.28	364.12	1.47	-3.72	0.22
7 km	$T=15$ s	15.71	0.89	0.344	0.070	6.46	0.40	-11.41	5.99	344.18	4.55	-34.80	0.91
	$T=30$ s	16.94	0.33	0.300	0.038	2.96	0.09	-7.01	2.39	449.64	7.14	-14.90	1.41
	$T=60$ s	17.93	0.13	0.363	0.005	1.50	0.01	-2.51	0.59	504.79	1.66	-4.52	0.29
PSH/C3													
3 km	$T=15$ s	83.49	9.06	0.945	0.036	1218.18	219.00	919.72	98.10	66.08	3.61	-70.87	1.97
	$T=30$ s	62.64	11.30	0.696	0.056	166.42	57.69	658.81	110.29	114.21	5.25	-49.70	2.00
	$T=60$ s	42.00	7.16	0.333	0.094	19.88	6.07	412.03	68.46	170.00	5.22	-25.19	1.76
5 km	$T=15$ s	124.08	16.34	0.943	0.105	1509.44	410.89	892.94	134.64	90.08	4.93	-76.16	1.50
	$T=30$ s	102.39	16.32	0.712	0.037	244.18	76.62	712.82	122.36	155.35	6.77	-58.89	1.87
	$T=60$ s	81.63	9.29	0.426	0.140	38.64	13.04	549.02	65.49	233.75	9.42	-38.16	2.31
7 km	$T=15$ s	161.81	16.10	0.975	0.156	1674.88	370.06	853.04	98.40	106.34	6.90	-79.89	1.33
	$T=30$ s	134.83	19.86	0.726	0.091	272.17	90.53	689.22	115.54	191.35	10.44	-63.81	2.00
	$T=60$ s	114.38	12.17	0.442	0.145	46.62	15.46	570.65	67.59	287.93	12.91	-45.57	2.36

Table C.10: State estimation performance summary statistics for the PSH model applied to targets moving in flat terrain (T1).

		$\mathbf{error_{xy}}$ (m)		$\mathbf{error_{s}}$ (m/s)		**divergence**		$\mathbf{\triangle error_{xy}}$ (%)		**area** (m^2)		$\mathbf{\triangle area}$ (%)	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
PSH/C4													
3 km	$T=15$ s	6.33	0.08	0.093	0.009	6.26	0.18	-27.23	1.23	93.46	3.16	-58.76	1.76
	$T=30$ s	6.92	0.13	0.075	0.002	2.71	0.04	-20.24	0.91	132.23	0.51	-41.80	0.25
	$T=60$ s	7.66	0.13	0.071	0.002	1.25	0.02	-10.81	0.74	173.42	0.59	-23.62	0.29
5 km	$T=15$ s	9.79	0.46	0.133	0.036	8.33	0.92	-26.35	4.28	135.61	9.71	-63.98	2.88
	$T=30$ s	9.97	0.13	0.092	0.009	3.23	0.10	-26.57	0.75	187.28	5.32	-50.35	1.62
	$T=60$ s	10.90	0.16	0.074	0.003	1.34	0.00	-18.92	1.27	250.32	1.24	-33.70	0.38
7 km	$T=15$ s	13.44	0.96	0.158	0.054	9.32	1.13	-24.89	6.03	172.16	15.44	-67.32	3.15
	$T=30$ s	13.44	0.47	0.120	0.026	3.77	0.31	-27.13	2.90	234.20	10.27	-55.62	2.11
	$T=60$ s	13.94	0.36	0.081	0.004	1.42	0.01	-24.51	1.40	314.82	1.83	-40.41	0.39

Continuation of Table C.10: State estimation performance summary statistics for the PSH model applied to targets moving in flat terrain (T1).

		$\mathbf{error_{xy}}$ (m)		$\mathbf{error_{s}}$ (m/s)		**divergence**		$\mathbf{\triangle error_{xy}}$ (%)		**area** (m^2)		$\mathbf{\triangle area}$ (%)	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
PSH/C1													
3 km	$T=15$ s	8.50	0.14	0.297	0.018	7.13	0.14	1.86	1.57	174.34	2.99	-23.20	1.45
	$T=30$ s	9.75	0.16	0.296	0.013	4.09	0.15	16.04	1.76	192.76	1.16	-15.15	0.62
	$T=60$ s	9.42	0.15	0.437	0.006	1.74	0.01	10.29	1.34	217.31	1.06	-4.29	0.28
5 km	$T=15$ s	12.60	0.38	0.342	0.031	7.47	0.19	-1.68	3.05	266.44	2.86	-29.38	0.81
	$T=30$ s	14.42	0.39	0.326	0.027	3.85	0.16	11.26	3.11	310.92	1.56	-17.61	0.45
	$T=60$ s	14.28	0.23	0.439	0.014	1.65	0.01	7.81	1.94	356.19	1.50	-5.60	0.32
7 km	$T=15$ s	16.45	0.66	0.368	0.037	7.57	0.34	-5.22	4.14	349.27	2.12	-33.84	0.38
	$T=30$ s	18.73	0.79	0.349	0.037	3.71	0.20	6.57	4.68	422.16	1.88	-20.03	0.31
	$T=60$ s	19.17	0.31	0.446	0.014	1.65	0.02	6.44	1.73	489.77	1.90	-7.22	0.32
PSH/C2													
3 km	$T=15$ s	7.85	0.12	0.234	0.017	5.63	0.17	-8.12	1.70	179.01	3.11	-21.16	1.50
	$T=30$ s	8.12	0.02	0.162	0.004	2.72	0.03	-5.76	0.33	198.57	0.98	-12.65	0.53
	$T=60$ s	8.52	0.02	0.327	0.004	1.50	0.02	-1.09	0.28	220.85	0.84	-2.80	0.21
5 km	$T=15$ s	11.65	0.26	0.299	0.021	6.10	0.15	-11.30	2.43	271.07	3.88	-28.16	1.09
	$T=30$ s	12.45	0.15	0.226	0.015	2.85	0.05	-6.81	1.19	324.80	3.17	-13.97	0.90
	$T=60$ s	13.21	0.08	0.345	0.003	1.49	0.01	-1.63	0.46	363.58	0.76	-3.69	0.12
7 km	$T=15$ s	15.22	0.48	0.338	0.022	6.33	0.19	-14.25	3.30	347.96	3.55	-34.09	0.70
	$T=30$ s	16.63	0.24	0.290	0.016	2.97	0.04	-8.47	1.54	440.60	2.22	-16.56	0.46
	$T=60$ s	17.89	0.11	0.368	0.006	1.52	0.01	-2.49	0.68	502.05	1.07	-4.92	0.17
PSH/C3													
3 km	$T=15$ s	68.93	9.08	0.880	0.069	793.04	262.04	750.34	113.63	66.41	6.93	-70.60	2.90
	$T=30$ s	51.69	7.34	0.653	0.130	104.71	33.33	536.68	89.58	113.95	7.76	-49.58	3.16
	$T=60$ s	37.09	3.26	0.361	0.081	14.95	3.08	355.56	38.86	169.07	4.98	-25.38	1.95
5 km	$T=15$ s	99.98	9.36	0.860	0.081	887.20	271.52	699.81	77.71	89.56	10.39	-76.22	2.71
	$T=30$ s	83.13	12.29	0.693	0.121	153.10	61.38	566.10	98.59	150.06	10.93	-60.18	2.87
	$T=60$ s	68.26	7.32	0.462	0.141	24.56	8.19	448.12	59.46	232.06	15.02	-38.47	3.95
7 km	$T=15$ s	123.36	8.95	0.843	0.080	875.95	256.00	627.59	56.40	110.35	10.25	-79.08	1.93
	$T=30$ s	107.54	12.76	0.682	0.141	164.28	69.61	533.63	74.81	184.76	17.33	-64.99	3.27
	$T=60$ s	94.91	9.40	0.496	0.156	30.08	10.86	461.29	54.95	282.18	23.19	-46.55	4.40

Table C.11: State estimation performance summary statistics for the PSH model applied to targets moving in rolling terrain (T2).

		$\mathbf{error_{xy}}$ (m)		$\mathbf{error_{s}}$ (m/s)		**divergence**		$\mathbf{\triangle error_{xy}}$ (%)		**area** (m^2)		$\mathbf{\triangle area}$ (%)	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
PSH/C4													
3 km	$T=15$ s	6.53	0.09	0.138	0.008	6.79	0.17	-24.41	1.30	96.71	2.88	-57.40	1.33
	$T=30$ s	7.08	0.06	0.126	0.003	3.29	0.06	-18.08	0.66	131.13	1.38	-42.21	0.66
	$T=60$ s	7.99	0.13	0.141	0.006	1.73	0.07	-6.80	1.30	172.40	1.28	-23.94	0.68
5 km	$T=15$ s	9.91	0.33	0.173	0.017	8.46	0.38	-25.48	2.89	140.80	6.04	-62.69	1.63
	$T=30$ s	10.23	0.07	0.144	0.005	3.79	0.10	-24.45	0.67	186.06	4.53	-50.71	1.21
	$T=60$ s	11.46	0.23	0.141	0.003	1.66	0.05	-14.74	1.61	250.30	4.26	-33.65	1.12
7 km	$T=15$ s	13.28	0.68	0.195	0.023	9.32	0.39	-25.92	4.47	179.20	8.97	-66.07	1.71
	$T=30$ s	13.65	0.40	0.168	0.014	4.24	0.16	-25.90	2.57	232.77	8.28	-55.93	1.58
	$T=60$ s	14.62	0.23	0.146	0.003	1.68	0.03	-21.02	1.31	312.74	6.43	-40.76	1.23

Continuation of Table C.11: State estimation performance summary statistics for the PSH model applied to targets moving in rolling terrain (T2).

		$\mathbf{error_{xy}}$ (m)		$\mathbf{error_{s}}$ (m/s)		**divergence**		$\mathbf{\triangle error_{xy}}$ (%)		**area** (m^2)		$\mathbf{\triangle area}$ (%)	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
PSH/C1													
3 km	$T=15$ s	8.28	0.23	0.301	0.031	6.85	0.19	-1.28	2.31	170.30	6.32	-25.13	2.03
	$T=30$ s	9.77	0.26	0.311	0.022	4.15	0.18	16.64	2.53	187.11	3.44	-17.79	0.66
	$T=60$ s	9.57	0.21	0.411	0.011	1.69	0.02	12.74	1.91	210.40	1.87	-7.67	0.21
5 km	$T=15$ s	12.30	0.63	0.347	0.056	7.31	0.44	-4.10	5.33	257.93	5.08	-31.66	1.13
	$T=30$ s	14.36	0.57	0.336	0.042	3.85	0.20	11.18	4.43	300.46	4.06	-20.41	0.79
	$T=60$ s	14.49	0.60	0.404	0.024	1.63	0.08	9.86	4.46	343.38	0.76	-9.07	0.27
7 km	$T=15$ s	16.16	1.13	0.372	0.069	7.50	0.63	-6.91	7.15	336.22	1.93	-36.31	0.44
	$T=30$ s	18.75	1.06	0.358	0.054	3.75	0.27	7.14	6.52	405.35	1.60	-23.21	0.43
	$T=60$ s	19.25	0.95	0.405	0.039	1.64	0.07	7.24	5.48	472.65	0.66	-10.49	0.28
PSH/C2													
3 km	$T=15$ s	7.80	0.21	0.247	0.026	5.51	0.20	-9.10	2.22	178.30	7.04	-21.72	2.39
	$T=30$ s	7.97	0.05	0.161	0.007	2.56	0.03	-7.69	0.27	194.56	2.97	-14.64	0.69
	$T=60$ s	8.37	0.08	0.276	0.002	1.37	0.01	-3.07	0.44	215.67	1.68	-5.39	0.24
5 km	$T=15$ s	11.47	0.42	0.312	0.031	5.98	0.24	-12.82	3.68	268.52	8.24	-28.89	1.92
	$T=30$ s	12.25	0.18	0.228	0.021	2.78	0.05	-8.26	1.40	317.14	4.74	-16.06	1.04
	$T=60$ s	12.99	0.08	0.301	0.005	1.43	0.01	-3.36	0.14	354.49	1.52	-6.17	0.30
7 km	$T=15$ s	14.87	0.71	0.349	0.031	6.21	0.23	-16.39	4.68	343.41	8.21	-34.96	1.41
	$T=30$ s	16.29	0.30	0.286	0.023	2.92	0.07	-10.04	1.88	428.09	2.90	-18.96	0.45
	$T=60$ s	17.62	0.19	0.330	0.005	1.48	0.01	-3.99	0.87	490.12	2.41	-7.22	0.40
PSH/C3													
3 km	$T=15$ s	60.97	16.89	0.780	0.096	509.89	302.55	648.62	206.61	68.32	8.54	-69.71	3.90
	$T=30$ s	38.08	7.37	0.518	0.085	57.33	17.50	363.82	90.35	115.71	6.70	-48.66	2.26
	$T=60$ s	24.43	2.55	0.289	0.023	8.16	0.81	195.34	31.63	165.95	9.87	-26.69	3.37
5 km	$T=15$ s	93.88	28.95	0.769	0.104	700.98	484.51	649.40	232.43	88.94	10.40	-76.36	2.78
	$T=30$ s	64.76	18.40	0.559	0.111	88.60	42.66	415.34	148.20	153.50	9.64	-59.15	2.39
	$T=60$ s	44.88	9.10	0.348	0.061	12.52	2.99	255.00	72.42	232.36	17.81	-38.28	4.36
7 km	$T=15$ s	117.28	33.95	0.757	0.091	730.57	482.74	590.24	200.86	109.16	12.47	-79.28	2.31
	$T=30$ s	91.16	28.76	0.566	0.119	111.86	64.60	435.08	170.05	185.47	12.71	-64.76	2.36
	$T=60$ s	64.26	16.12	0.361	0.082	14.32	4.85	275.51	94.22	286.30	27.56	-45.67	5.08

Table C.12: State estimation performance summary statistics for the PSH model applied to targets moving in mountain terrain (T3).

		$\mathbf{error_{xy}}$ (m)		$\mathbf{error_{s}}$ (m/s)		**divergence**		$\mathbf{\triangle error_{xy}}$ (%)		**area** (m^2)		$\mathbf{\triangle area}$ (%)	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
PSH/C4													
3 km	$T=15$ s	6.59	0.14	0.143	0.010	6.22	0.15	-23.97	1.48	102.90	3.92	-54.81	1.31
	$T=30$ s	6.95	0.04	0.110	0.002	2.88	0.04	-19.58	0.58	132.88	2.87	-41.38	0.69
	$T=60$ s	7.78	0.15	0.117	0.005	1.57	0.08	-9.61	1.43	172.26	3.90	-24.07	0.72
5 km	$T=15$ s	9.88	0.57	0.180	0.025	7.58	0.52	-25.72	4.98	149.83	8.11	-60.39	2.07
	$T=30$ s	10.20	0.11	0.135	0.008	3.45	0.11	-24.56	1.11	191.38	6.18	-49.26	1.47
	$T=60$ s	11.25	0.16	0.121	0.008	1.57	0.08	-16.53	1.09	251.51	5.81	-33.28	1.20
7 km	$T=15$ s	13.09	1.12	0.203	0.035	8.32	0.62	-27.01	7.13	189.62	12.57	-64.15	2.38
	$T=30$ s	13.52	0.50	0.159	0.017	3.87	0.21	-26.29	3.37	238.97	8.67	-54.73	1.56
	$T=60$ s	14.58	0.25	0.131	0.010	1.61	0.06	-21.40	1.10	318.33	8.70	-39.68	1.53

Continuation of Table C.12: State estimation performance summary statistics for the PSH model applied to targets moving in mountain terrain (T3).

		$\mathbf{err}_{\mu_x}$ (m)	$\mathbf{err}_{\mu_y}$ (m)	$\mathbf{err}_{\mu_s}$ (m/s)	$\mathbf{err}_{\sigma_x}$ (m)	$\mathbf{err}_{\sigma_y}$ (m)	$\mathbf{err}_{\sigma_s}$ (m/s)	$\mathbf{rej}_{\mu_x}$ (%)	$\mathbf{rej}_{\mu_y}$ (%)	$\mathbf{rej}_{\mu_s}$ (%)	$\mathbf{rej}_{\sigma_x}$ (%)	$\mathbf{rej}_{\sigma_y}$ (%)	$\mathbf{rej}_{\sigma_s}$ (%)
PSH-ExpCorr/C1													
3 km	$T=15$ s	0.04	-0.03	0.04	0.29	0.12	-0.03	0.39	0.59	0.52	0.05	0.03	0.63
	$T=30$ s	0.09	0.03	0.02	0.10	-0.04	-0.09	0.14	0.16	0.76	0.05	0.05	0.92
	$T=60$ s	-0.01	0.09	0.01	-0.20	-0.30	-0.13	0.23	0.26	0.95	0.03	0.05	0.96
5 km	$T=15$ s	0.12	-0.01	0.07	0.54	0.35	0.00	0.52	0.69	0.79	0.09	0.07	0.70
	$T=30$ s	0.17	-0.01	0.03	0.68	0.45	-0.05	0.28	0.46	0.62	0.12	0.11	0.81
	$T=60$ s	0.11	0.14	0.01	0.18	0.06	-0.14	0.24	0.22	0.89	0.09	0.10	0.93
7 km	$T=15$ s	0.06	-0.09	0.09	0.75	0.49	-0.00	0.55	0.71	0.90	0.12	0.10	0.69
	$T=30$ s	0.10	-0.15	0.05	1.25	0.98	-0.03	0.40	0.61	0.68	0.16	0.13	0.81
	$T=60$ s	0.14	0.02	0.02	0.82	0.54	-0.14	0.19	0.27	0.81	0.12	0.15	0.92
PSH-ExpCorr/C2													
3 km	$T=15$ s	0.04	-0.02	0.03	0.27	0.10	-0.03	0.41	0.60	0.51	0.04	0.03	0.63
	$T=30$ s	0.10	0.02	0.02	0.08	-0.06	-0.09	0.15	0.17	0.77	0.05	0.05	0.92
	$T=60$ s	0.01	0.11	0.01	-0.20	-0.30	-0.13	0.22	0.24	0.94	0.03	0.05	0.96
5 km	$T=15$ s	0.13	-0.00	0.06	0.52	0.34	0.00	0.53	0.70	0.77	0.09	0.06	0.69
	$T=30$ s	0.18	-0.01	0.03	0.64	0.41	-0.05	0.29	0.48	0.61	0.11	0.10	0.81
	$T=60$ s	0.14	0.16	0.01	0.17	0.06	-0.14	0.23	0.25	0.89	0.09	0.10	0.93
7 km	$T=15$ s	0.08	-0.08	0.08	0.73	0.48	-0.01	0.56	0.72	0.89	0.12	0.10	0.67
	$T=30$ s	0.13	-0.15	0.05	1.20	0.95	-0.03	0.40	0.62	0.65	0.15	0.12	0.81
	$T=60$ s	0.16	0.04	0.01	0.79	0.51	-0.14	0.19	0.29	0.82	0.13	0.14	0.92
PSH-ExpCorr/C3													
3 km	$T=15$ s	1.17	0.04	0.07	0.40	0.12	0.00	0.49	0.65	0.79	0.14	0.11	0.64
	$T=30$ s	0.41	-0.08	0.03	-0.09	-0.11	-0.10	0.31	0.36	0.84	0.03	0.08	0.95
	$T=60$ s	-0.03	-0.29	0.02	-0.36	-0.40	-0.17	0.27	0.29	0.94	0.02	0.05	0.98
5 km	$T=15$ s	1.50	0.04	0.11	0.56	0.34	0.02	0.58	0.72	0.93	0.12	0.12	0.61
	$T=30$ s	0.58	-0.15	0.04	0.26	0.19	-0.08	0.38	0.56	0.83	0.05	0.10	0.82
	$T=60$ s	0.21	-0.30	0.02	-0.12	-0.22	-0.17	0.30	0.39	0.87	0.05	0.05	0.94
7 km	$T=15$ s	1.62	-0.05	0.14	0.73	0.43	0.03	0.59	0.71	0.97	0.13	0.13	0.59
	$T=30$ s	0.70	-0.25	0.06	0.67	0.52	-0.07	0.49	0.65	0.76	0.08	0.10	0.79
	$T=60$ s	0.19	-0.47	0.02	0.48	0.10	-0.18	0.24	0.45	0.83	0.07	0.11	0.93

Table C.13: Consistency summary statistics for the PSH-ExpCorr model applied to targets moving in flat terrain (T1).

		$\mathbf{err}_{\mu_x}$ (m)	$\mathbf{err}_{\mu_y}$ (m)	$\mathbf{err}_{\mu_s}$ (m/s)	$\mathbf{err}_{\sigma_x}$ (m)	$\mathbf{err}_{\sigma_y}$ (m)	$\mathbf{err}_{\sigma_s}$ (m/s)	$\mathbf{rej}_{\mu_x}$ (%)	$\mathbf{rej}_{\mu_y}$ (%)	$\mathbf{rej}_{\mu_s}$ (%)	$\mathbf{rej}_{\sigma_x}$ (%)	$\mathbf{rej}_{\sigma_y}$ (%)	$\mathbf{rej}_{\sigma_s}$ (%)
PSH-ExpCorr/C1													
3 km	$T=15$ s	-0.02	0.04	0.05	0.26	0.13	-0.01	0.46	0.54	0.62	0.05	0.04	0.59
	$T=30$ s	-0.03	0.01	0.05	0.22	0.09	-0.08	0.23	0.19	0.79	0.07	0.06	0.86
	$T=60$ s	-0.23	0.34	0.07	0.07	0.01	-0.13	0.31	0.37	0.96	0.07	0.07	0.94
5 km	$T=15$ s	0.02	-0.01	0.08	0.53	0.35	0.01	0.64	0.75	0.77	0.10	0.06	0.66
	$T=30$ s	-0.05	-0.02	0.06	0.85	0.69	-0.03	0.27	0.40	0.65	0.14	0.12	0.75
	$T=60$ s	-0.30	0.37	0.07	0.74	0.65	-0.12	0.23	0.28	0.93	0.16	0.12	0.89
7 km	$T=15$ s	0.02	0.07	0.10	0.71	0.47	0.01	0.71	0.86	0.85	0.13	0.09	0.66
	$T=30$ s	-0.12	0.03	0.08	1.42	1.24	-0.01	0.48	0.65	0.62	0.20	0.16	0.80
	$T=60$ s	-0.26	0.34	0.07	1.53	1.45	-0.12	0.22	0.14	0.82	0.25	0.23	0.84
PSH-ExpCorr/C2													
3 km	$T=15$ s	-0.01	0.05	0.04	0.25	0.11	-0.01	0.47	0.57	0.59	0.05	0.04	0.59
	$T=30$ s	-0.03	0.00	0.04	0.20	0.08	-0.08	0.23	0.20	0.79	0.07	0.05	0.87
	$T=60$ s	-0.22	0.35	0.06	0.06	0.00	-0.13	0.29	0.37	0.95	0.07	0.07	0.94
5 km	$T=15$ s	0.02	-0.02	0.07	0.51	0.33	0.01	0.66	0.77	0.75	0.09	0.06	0.65
	$T=30$ s	-0.05	-0.03	0.06	0.81	0.66	-0.03	0.30	0.41	0.63	0.14	0.12	0.76
	$T=60$ s	-0.29	0.39	0.07	0.73	0.63	-0.12	0.23	0.29	0.94	0.16	0.12	0.89
7 km	$T=15$ s	0.03	0.07	0.09	0.68	0.45	0.00	0.72	0.87	0.86	0.12	0.08	0.65
	$T=30$ s	-0.10	0.02	0.07	1.37	1.20	-0.02	0.50	0.66	0.59	0.20	0.15	0.79
	$T=60$ s	-0.26	0.36	0.07	1.51	1.43	-0.12	0.23	0.14	0.83	0.23	0.22	0.84
PSH-ExpCorr/C3													
3 km	$T=15$ s	0.67	-0.07	0.10	0.19	0.01	-0.00	0.67	0.60	0.75	0.09	0.06	0.60
	$T=30$ s	0.07	-0.09	0.06	-0.04	-0.05	-0.10	0.63	0.56	0.87	0.03	0.04	0.87
	$T=60$ s	-0.48	0.40	0.08	-0.12	-0.19	-0.15	0.67	0.58	0.95	0.04	0.05	0.94
5 km	$T=15$ s	1.05	-0.09	0.13	0.38	0.26	0.02	0.78	0.75	0.87	0.10	0.08	0.60
	$T=30$ s	0.09	-0.16	0.08	0.41	0.35	-0.07	0.58	0.52	0.73	0.06	0.07	0.78
	$T=60$ s	-0.68	0.38	0.08	0.41	0.30	-0.15	0.57	0.58	0.94	0.08	0.07	0.90
7 km	$T=15$ s	1.26	-0.04	0.16	0.56	0.41	0.03	0.78	0.80	0.90	0.10	0.09	0.59
	$T=30$ s	0.12	-0.09	0.09	0.83	0.78	-0.07	0.65	0.67	0.75	0.09	0.10	0.77
	$T=60$ s	-0.80	0.29	0.08	1.00	0.91	-0.15	0.56	0.53	0.89	0.14	0.13	0.88

Table C.14: Consistency summary statistics for the PSH-ExpCorr model applied to targets moving in rolling terrain (T2).

		$\mathbf{err}_{\mu_x}$ (m)	$\mathbf{err}_{\mu_y}$ (m)	$\mathbf{err}_{\mu_s}$ (m/s)	$\mathbf{err}_{\sigma_x}$ (m)	$\mathbf{err}_{\sigma_y}$ (m)	$\mathbf{err}_{\sigma_s}$ (m/s)	$\mathbf{rej}_{\mu_x}$ (%)	$\mathbf{rej}_{\mu_y}$ (%)	$\mathbf{rej}_{\mu_s}$ (%)	$\mathbf{rej}_{\sigma_x}$ (%)	$\mathbf{rej}_{\sigma_y}$ (%)	$\mathbf{rej}_{\sigma_s}$ (%)
PSH-ExpCorr/C1													
3 km	$T=15$ s	-0.04	-0.02	0.05	0.05	0.14	-0.03	0.55	0.39	0.71	0.03	0.03	0.55
	$T=30$ s	-0.06	-0.01	0.06	-0.03	0.09	-0.11	0.17	0.13	0.75	0.05	0.04	0.78
	$T=60$ s	-0.18	0.11	0.09	-0.08	-0.04	-0.16	0.27	0.23	0.95	0.05	0.04	0.94
5 km	$T=15$ s	-0.00	-0.03	0.07	0.21	0.35	-0.01	0.72	0.58	0.86	0.05	0.07	0.55
	$T=30$ s	-0.09	0.03	0.07	0.46	0.61	-0.06	0.44	0.28	0.70	0.08	0.10	0.69
	$T=60$ s	-0.19	0.03	0.09	0.40	0.52	-0.15	0.22	0.17	0.89	0.10	0.11	0.89
7 km	$T=15$ s	-0.04	-0.06	0.08	0.31	0.48	-0.02	0.74	0.61	0.93	0.07	0.09	0.52
	$T=30$ s	-0.07	-0.09	0.08	0.82	0.99	-0.04	0.62	0.47	0.74	0.12	0.14	0.71
	$T=60$ s	-0.23	-0.03	0.09	1.05	1.30	-0.14	0.18	0.10	0.83	0.17	0.20	0.78
PSH-ExpCorr/C2													
3 km	$T=15$ s	-0.04	-0.02	0.03	0.04	0.14	-0.03	0.57	0.40	0.69	0.02	0.03	0.55
	$T=30$ s	-0.06	-0.01	0.05	-0.05	0.08	-0.10	0.17	0.13	0.74	0.04	0.03	0.77
	$T=60$ s	-0.15	0.11	0.08	-0.08	-0.05	-0.16	0.27	0.22	0.93	0.05	0.04	0.94
5 km	$T=15$ s	-0.00	-0.03	0.06	0.21	0.35	-0.02	0.73	0.60	0.84	0.05	0.07	0.54
	$T=30$ s	-0.08	0.03	0.06	0.44	0.59	-0.06	0.45	0.29	0.69	0.07	0.10	0.69
	$T=60$ s	-0.17	0.03	0.08	0.37	0.49	-0.15	0.22	0.15	0.87	0.10	0.10	0.88
7 km	$T=15$ s	-0.03	-0.05	0.07	0.31	0.48	-0.03	0.75	0.63	0.91	0.07	0.09	0.51
	$T=30$ s	-0.06	-0.09	0.08	0.80	0.96	-0.04	0.64	0.49	0.75	0.12	0.13	0.71
	$T=60$ s	-0.21	-0.03	0.09	1.01	1.25	-0.14	0.18	0.10	0.82	0.16	0.19	0.78
PSH-ExpCorr/C3													
3 km	$T=15$ s	1.23	-0.02	0.10	0.20	0.08	-0.02	0.75	0.57	0.83	0.13	0.08	0.51
	$T=30$ s	0.28	0.03	0.07	-0.25	-0.08	-0.12	0.55	0.52	0.83	0.03	0.05	0.79
	$T=60$ s	-0.11	0.05	0.10	-0.19	-0.17	-0.20	0.59	0.51	0.95	0.05	0.04	0.94
5 km	$T=15$ s	1.62	-0.00	0.13	0.32	0.31	0.00	0.81	0.62	0.93	0.13	0.10	0.49
	$T=30$ s	0.38	0.04	0.08	0.02	0.24	-0.10	0.65	0.57	0.78	0.03	0.06	0.72
	$T=60$ s	-0.10	-0.03	0.10	0.14	0.18	-0.19	0.52	0.44	0.90	0.08	0.07	0.87
7 km	$T=15$ s	1.84	-0.05	0.15	0.40	0.50	0.00	0.82	0.64	0.96	0.13	0.12	0.47
	$T=30$ s	0.54	-0.09	0.09	0.29	0.57	-0.09	0.72	0.58	0.80	0.05	0.08	0.67
	$T=60$ s	-0.14	-0.03	0.10	0.56	0.74	-0.19	0.49	0.44	0.85	0.10	0.11	0.82

Table C.15: Consistency summary statistics for the PSH-ExpCorr model applied to targets moving in mountain terrain (T3).

		$\mathbf{error_{xy}}$ (m)		$\mathbf{error_{s}}$ (m/s)		**divergence**		$\mathbf{\triangle error_{xy}}$ (%)		**area** (m^2)		$\mathbf{\triangle area}$ (%)	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
PSH-ExpCorr/C1													
3 km	$T=15$ s	6.74	0.09	0.106	0.011	5.72	0.21	-21.83	1.27	122.88	2.94	-45.82	1.69
	$T=30$ s	7.07	0.14	0.075	0.002	2.57	0.04	-18.08	1.01	146.27	1.14	-35.69	0.16
	$T=60$ s	7.66	0.17	0.061	0.003	1.21	0.02	-10.33	1.13	170.07	1.86	-25.15	0.32
5 km	$T=15$ s	10.24	0.47	0.154	0.036	7.10	0.69	-22.26	4.52	179.18	8.90	-52.43	2.66
	$T=30$ s	10.25	0.16	0.099	0.008	2.97	0.05	-23.95	0.82	213.26	5.52	-43.49	1.66
	$T=60$ s	10.85	0.22	0.068	0.005	1.31	0.01	-18.77	1.77	247.31	3.05	-34.54	0.61
7 km	$T=15$ s	13.96	1.06	0.187	0.057	7.67	0.83	-21.36	6.82	225.47	13.66	-57.23	2.80
	$T=30$ s	13.70	0.42	0.132	0.021	3.33	0.20	-25.12	2.54	273.90	12.56	-48.11	2.56
	$T=60$ s	13.85	0.41	0.080	0.007	1.39	0.02	-24.44	1.81	315.56	3.12	-40.31	0.50
PSH-ExpCorr/C2													
3 km	$T=15$ s	6.74	0.09	0.104	0.011	5.65	0.20	-21.81	1.34	122.97	2.86	-45.78	1.65
	$T=30$ s	7.07	0.14	0.075	0.002	2.55	0.03	-18.18	1.01	146.72	1.16	-35.50	0.18
	$T=60$ s	7.66	0.16	0.060	0.003	1.21	0.02	-10.37	1.14	170.18	1.80	-25.11	0.30
5 km	$T=15$ s	10.30	0.51	0.150	0.035	7.02	0.64	-21.70	4.88	178.75	8.77	-52.54	2.62
	$T=30$ s	10.24	0.15	0.098	0.007	2.94	0.05	-24.02	0.78	213.78	5.21	-43.36	1.57
	$T=60$ s	10.84	0.22	0.067	0.005	1.31	0.01	-18.86	1.74	247.48	2.98	-34.50	0.60
7 km	$T=15$ s	14.13	1.16	0.183	0.055	7.62	0.80	-20.28	7.45	224.32	13.19	-57.45	2.71
	$T=30$ s	13.73	0.41	0.130	0.020	3.31	0.19	-24.93	2.51	274.62	12.21	-47.98	2.49
	$T=60$ s	13.83	0.39	0.079	0.007	1.38	0.02	-24.53	1.69	315.80	3.02	-40.27	0.48
PSH-ExpCorr/C3													
3 km	$T=15$ s	7.46	0.19	0.136	0.007	8.54	1.02	-10.94	2.66	123.95	2.86	-45.38	1.64
	$T=30$ s	7.32	0.15	0.080	0.004	2.73	0.13	-13.74	1.41	147.07	1.56	-35.35	0.46
	$T=60$ s	7.98	0.10	0.064	0.002	1.28	0.04	-5.64	1.26	170.46	1.73	-24.91	0.29
5 km	$T=15$ s	11.13	0.39	0.179	0.029	9.40	0.69	-12.80	3.70	181.16	7.68	-51.92	2.31
	$T=30$ s	10.52	0.17	0.100	0.006	3.03	0.08	-19.74	1.27	216.36	4.86	-42.71	1.41
	$T=60$ s	11.21	0.15	0.070	0.002	1.37	0.04	-14.69	0.81	248.47	3.41	-34.21	0.73
7 km	$T=15$ s	15.04	0.99	0.214	0.048	9.80	0.86	-12.73	6.28	228.40	11.02	-56.68	2.28
	$T=30$ s	13.82	0.36	0.128	0.017	3.27	0.11	-22.15	2.19	278.55	12.74	-47.24	2.59
	$T=60$ s	14.20	0.39	0.080	0.004	1.42	0.02	-21.16	1.44	318.39	3.84	-39.77	0.64

Table C.16: State estimation performance summary statistics for the PSH-ExpCorr model applied to targets moving in flat terrain (T1).

		$\mathbf{error_{xy}}$ (m)		$\mathbf{error_{s}}$ (m/s)		**divergence**		$\mathbf{\triangle error_{xy}}$ (%)		**area** (m^2)		$\mathbf{\triangle area}$ (%)	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
PSH-ExpCorr/C1													
3 km	$T=15$ s	6.88	0.09	0.145	0.010	6.10	0.19	-19.73	1.31	126.33	2.81	-44.34	1.32
	$T=30$ s	7.23	0.05	0.120	0.003	3.16	0.06	-15.99	0.50	146.79	0.95	-35.34	0.53
	$T=60$ s	8.16	0.16	0.131	0.004	1.82	0.04	-4.41	1.60	170.80	1.33	-24.73	0.69
5 km	$T=15$ s	10.29	0.31	0.188	0.018	7.32	0.32	-21.97	2.80	183.72	5.68	-51.30	1.55
	$T=30$ s	10.50	0.08	0.146	0.006	3.43	0.09	-21.88	0.73	215.22	4.37	-42.99	1.20
	$T=60$ s	11.62	0.22	0.139	0.008	1.70	0.04	-12.98	1.54	250.93	3.12	-33.51	0.84
7 km	$T=15$ s	13.71	0.67	0.215	0.024	7.77	0.33	-23.05	4.48	231.67	8.43	-56.13	1.62
	$T=30$ s	13.93	0.36	0.174	0.013	3.66	0.10	-23.74	2.32	276.60	8.08	-47.64	1.56
	$T=60$ s	14.73	0.21	0.150	0.007	1.69	0.03	-19.72	1.28	317.79	4.18	-39.82	0.82
PSH-ExpCorr/C2													
3 km	$T=15$ s	6.89	0.09	0.142	0.010	6.01	0.18	-19.57	1.37	126.61	2.77	-44.21	1.30
	$T=30$ s	7.24	0.06	0.119	0.003	3.12	0.06	-15.91	0.55	147.70	0.84	-34.94	0.48
	$T=60$ s	8.15	0.16	0.130	0.004	1.81	0.04	-4.53	1.67	170.97	1.43	-24.65	0.73
5 km	$T=15$ s	10.36	0.33	0.184	0.017	7.24	0.31	-21.35	2.98	183.45	5.53	-51.38	1.51
	$T=30$ s	10.51	0.07	0.144	0.006	3.38	0.09	-21.73	0.69	216.42	4.05	-42.68	1.11
	$T=60$ s	11.60	0.23	0.138	0.008	1.69	0.04	-13.12	1.58	251.29	3.29	-33.41	0.89
7 km	$T=15$ s	13.86	0.73	0.210	0.023	7.72	0.32	-22.08	4.85	230.56	8.03	-56.34	1.55
	$T=30$ s	13.98	0.37	0.171	0.013	3.62	0.10	-23.41	2.35	277.82	7.82	-47.41	1.51
	$T=60$ s	14.72	0.21	0.149	0.007	1.68	0.03	-19.81	1.26	318.31	4.42	-39.72	0.86
PSH-ExpCorr/C3													
3 km	$T=15$ s	7.66	0.18	0.173	0.014	8.57	0.88	-7.85	2.06	127.85	2.23	-43.67	1.05
	$T=30$ s	8.02	0.04	0.135	0.007	3.68	0.15	-4.91	0.48	148.68	1.49	-34.53	0.78
	$T=60$ s	8.69	0.08	0.146	0.002	1.92	0.08	3.20	1.32	170.11	1.87	-25.01	0.92
5 km	$T=15$ s	11.38	0.41	0.215	0.022	9.82	1.17	-10.82	3.12	185.72	4.88	-50.79	1.34
	$T=30$ s	11.45	0.21	0.157	0.010	3.87	0.23	-11.83	1.56	219.55	5.20	-41.86	1.39
	$T=60$ s	12.39	0.16	0.150	0.006	1.79	0.09	-5.38	1.22	252.34	2.74	-33.13	0.75
7 km	$T=15$ s	15.03	0.81	0.246	0.028	10.23	1.25	-12.85	4.98	234.29	6.52	-55.64	1.26
	$T=30$ s	14.84	0.50	0.178	0.016	3.89	0.20	-15.79	2.89	282.44	8.29	-46.54	1.60
	$T=60$ s	15.60	0.34	0.159	0.009	1.77	0.11	-12.80	2.13	322.59	5.26	-38.92	1.01

Table C.17: State estimation performance summary statistics for the PSH-ExpCorr model applied to targets moving in rolling terrain (T2).

		$\mathbf{error_{xy}}$ (m)		$\mathbf{error_{s}}$ (m/s)		**divergence**		$\mathbf{\triangle error_{xy}}$ (%)		**area** (m^2)		$\mathbf{\triangle area}$ (%)	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
PSH-ExpCorr/C1													
3 km	$T=15$ s	6.97	0.14	0.153	0.016	5.79	0.21	-19.08	1.49	131.05	4.37	-42.45	1.40
	$T=30$ s	7.12	0.04	0.111	0.002	2.80	0.03	-17.36	0.79	147.25	2.05	-35.11	0.56
	$T=60$ s	7.87	0.14	0.128	0.005	1.67	0.07	-8.32	1.57	171.40	2.99	-24.46	0.36
5 km	$T=15$ s	10.35	0.54	0.202	0.032	6.83	0.48	-21.60	4.76	190.35	8.47	-49.64	2.11
	$T=30$ s	10.58	0.10	0.147	0.011	3.21	0.10	-21.22	0.91	221.65	5.33	-41.33	1.25
	$T=60$ s	11.34	0.19	0.136	0.005	1.60	0.06	-15.34	1.24	252.88	3.64	-32.92	0.63
7 km	$T=15$ s	13.64	1.10	0.233	0.042	7.29	0.52	-23.50	7.04	238.66	12.54	-54.86	2.34
	$T=30$ s	14.02	0.43	0.178	0.020	3.45	0.17	-23.05	2.94	286.41	8.64	-45.83	1.59
	$T=60$ s	14.72	0.27	0.150	0.008	1.62	0.04	-20.02	1.10	325.45	6.30	-38.34	1.09
PSH-ExpCorr/C2													
3 km	$T=15$ s	6.99	0.15	0.157	0.015	5.82	0.21	-18.80	1.61	131.72	4.39	-42.17	1.39
	$T=30$ s	7.14	0.05	0.112	0.003	2.78	0.03	-17.21	0.84	148.76	1.95	-34.50	0.56
	$T=60$ s	7.83	0.14	0.127	0.005	1.66	0.07	-8.74	1.51	171.61	2.87	-24.37	0.31
5 km	$T=15$ s	10.39	0.58	0.203	0.030	6.82	0.46	-21.18	5.06	189.80	8.41	-49.78	2.09
	$T=30$ s	10.60	0.10	0.147	0.010	3.20	0.09	-21.11	0.94	222.82	5.19	-41.03	1.22
	$T=60$ s	11.28	0.19	0.135	0.005	1.59	0.06	-15.78	1.26	253.25	3.63	-32.84	0.63
7 km	$T=15$ s	13.75	1.18	0.232	0.040	7.30	0.52	-22.77	7.52	236.96	12.11	-55.17	2.25
	$T=30$ s	14.04	0.45	0.178	0.019	3.43	0.16	-22.90	3.07	286.94	8.64	-45.73	1.59
	$T=60$ s	14.67	0.27	0.148	0.008	1.61	0.04	-20.35	1.11	325.91	6.12	-38.26	1.06
PSH-ExpCorr/C3													
3 km	$T=15$ s	7.97	0.12	0.196	0.012	8.96	0.64	-4.29	1.46	129.84	4.56	-42.92	1.40
	$T=30$ s	7.85	0.09	0.135	0.011	3.31	0.21	-6.76	1.22	150.07	2.35	-33.98	0.30
	$T=60$ s	8.22	0.17	0.143	0.005	1.73	0.06	-3.07	2.16	171.54	2.71	-24.43	0.28
5 km	$T=15$ s	11.68	0.40	0.239	0.024	9.97	0.81	-8.30	3.10	187.56	8.28	-50.34	2.02
	$T=30$ s	11.32	0.09	0.161	0.015	3.65	0.18	-12.76	1.03	223.81	5.69	-40.79	1.32
	$T=60$ s	11.80	0.21	0.150	0.006	1.68	0.08	-10.30	1.54	255.46	4.05	-32.27	0.76
7 km	$T=15$ s	15.28	0.88	0.269	0.033	10.29	0.95	-11.40	5.28	234.90	11.67	-55.54	2.15
	$T=30$ s	14.72	0.42	0.184	0.022	3.69	0.17	-16.24	2.61	288.40	10.68	-45.45	1.97
	$T=60$ s	15.35	0.21	0.160	0.007	1.68	0.05	-14.61	0.98	332.56	8.45	-37.03	1.53

Table C.18: State estimation performance summary statistics for the PSH-ExpCorr model applied to targets moving in mountain terrain (T3).

		$\mathbf{err}_{\mu_x}$ (m)	$\mathbf{err}_{\mu_y}$ (m)	$\mathbf{err}_{\mu_s}$ (m/s)	$\mathbf{err}_{\sigma_x}$ (m)	$\mathbf{err}_{\sigma_y}$ (m)	$\mathbf{err}_{\sigma_s}$ (m/s)	$\mathbf{rej}_{\mu_x}$ (%)	$\mathbf{rej}_{\mu_y}$ (%)	$\mathbf{rej}_{\mu_s}$ (%)	$\mathbf{rej}_{\sigma_x}$ (%)	$\mathbf{rej}_{\sigma_y}$ (%)	$\mathbf{rej}_{\sigma_s}$ (%)
PSH-Road/C1													
3 km	$T=15$ s	0.00	-0.01	0.06	-1.00	-0.97	-0.18	0.13	0.15	0.29	0.26	0.21	0.66
	$T=30$ s	0.02	-0.01	0.03	-1.02	-1.01	-0.24	0.05	0.08	0.26	0.31	0.24	0.97
	$T=60$ s	0.05	-0.01	0.03	-0.99	-0.97	-0.37	0.19	0.24	0.37	0.27	0.26	1.00
5 km	$T=15$ s	0.05	0.01	0.11	-1.66	-1.58	-0.19	0.41	0.57	0.87	0.48	0.38	0.66
	$T=30$ s	0.04	-0.02	0.04	-1.70	-1.61	-0.23	0.12	0.19	0.32	0.53	0.41	0.85
	$T=60$ s	0.18	-0.05	0.03	-1.68	-1.58	-0.40	0.30	0.27	0.28	0.47	0.39	1.00
7 km	$T=15$ s	0.01	-0.07	0.14	-2.17	-2.06	-0.20	0.53	0.72	0.96	0.57	0.49	0.64
	$T=30$ s	0.01	-0.01	0.06	-2.19	-2.11	-0.23	0.20	0.31	0.47	0.58	0.53	0.83
	$T=60$ s	0.17	-0.13	0.03	-2.19	-2.17	-0.42	0.32	0.37	0.27	0.54	0.54	0.93
PSH-Road/C2													
3 km	$T=15$ s	0.09	-0.01	0.04	-0.96	-0.97	-0.22	0.30	0.47	0.49	0.32	0.32	0.72
	$T=30$ s	0.18	0.22	0.03	-0.93	-0.97	-0.23	0.31	0.35	0.40	0.44	0.41	0.94
	$T=60$ s	-0.07	0.29	0.05	-1.10	-1.05	-0.25	0.26	0.35	0.56	0.47	0.39	0.99
5 km	$T=15$ s	0.14	0.00	0.08	-1.50	-1.47	-0.21	0.45	0.63	0.85	0.52	0.48	0.66
	$T=30$ s	0.33	0.10	0.03	-0.79	-1.16	-0.24	0.28	0.39	0.45	0.54	0.54	0.85
	$T=60$ s	-0.01	0.35	0.04	-1.68	-1.58	-0.27	0.37	0.47	0.54	0.67	0.64	0.98
7 km	$T=15$ s	0.14	-0.04	0.10	-1.69	-1.68	-0.21	0.54	0.70	0.92	0.58	0.59	0.62
	$T=30$ s	0.17	0.21	0.04	-1.42	-1.81	-0.23	0.30	0.49	0.62	0.57	0.55	0.81
	$T=60$ s	-0.04	0.45	0.04	-1.91	-1.91	-0.28	0.40	0.47	0.68	0.64	0.64	0.95

Table C.19: Consistency summary statistics for the PSH-Road model applied to targets moving in flat terrain (T1).

		$\mathbf{err}_{\mu_x}$ (m)	$\mathbf{err}_{\mu_y}$ (m)	$\mathbf{err}_{\mu_s}$ (m/s)	$\mathbf{err}_{\sigma_x}$ (m)	$\mathbf{err}_{\sigma_y}$ (m)	$\mathbf{err}_{\sigma_s}$ (m/s)	$\mathbf{rej}_{\mu_x}$ (%)	$\mathbf{rej}_{\mu_y}$ (%)	$\mathbf{rej}_{\mu_s}$ (%)	$\mathbf{rej}_{\sigma_x}$ (%)	$\mathbf{rej}_{\sigma_y}$ (%)	$\mathbf{rej}_{\sigma_s}$ (%)
PSH-Road/C1													
3 km	$T=15$ s	0.01	0.02	0.09	-0.86	-0.84	-0.17	0.22	0.29	0.42	0.16	0.12	0.54
	$T=30$ s	0.06	-0.02	0.06	-0.88	-0.86	-0.24	0.17	0.21	0.35	0.19	0.14	0.82
	$T=60$ s	-0.02	-0.01	0.06	-0.92	-0.82	-0.41	0.30	0.15	0.48	0.18	0.12	0.95
5 km	$T=15$ s	0.00	-0.00	0.13	-1.27	-1.21	-0.18	0.52	0.66	0.74	0.33	0.24	0.52
	$T=30$ s	0.04	-0.12	0.07	-1.37	-1.29	-0.23	0.25	0.28	0.39	0.39	0.28	0.76
	$T=60$ s	0.10	-0.08	0.06	-1.43	-1.15	-0.42	0.43	0.28	0.44	0.35	0.25	0.84
7 km	$T=15$ s	-0.09	-0.02	0.15	-1.27	-1.16	-0.18	0.67	0.79	0.87	0.39	0.31	0.50
	$T=30$ s	0.01	-0.19	0.09	-1.45	-1.37	-0.24	0.36	0.41	0.47	0.43	0.35	0.71
	$T=60$ s	0.05	-0.22	0.08	-1.81	-1.40	-0.42	0.47	0.31	0.44	0.43	0.35	0.86
PSH-Road/C2													
3 km	$T=15$ s	0.10	0.57	0.07	-0.37	-0.16	-0.19	0.44	0.50	0.57	0.27	0.23	0.62
	$T=30$ s	0.01	0.19	0.06	-0.50	-0.20	-0.22	0.48	0.57	0.60	0.39	0.33	0.86
	$T=60$ s	-0.11	0.47	0.13	-0.74	-0.46	-0.23	0.46	0.51	0.75	0.38	0.38	0.91
5 km	$T=15$ s	0.11	1.08	0.11	-0.07	0.50	-0.17	0.58	0.67	0.80	0.40	0.37	0.56
	$T=30$ s	0.05	0.39	0.07	-0.18	0.43	-0.21	0.43	0.50	0.55	0.49	0.46	0.75
	$T=60$ s	0.03	0.97	0.11	-0.26	0.57	-0.23	0.51	0.56	0.77	0.56	0.54	0.86
7 km	$T=15$ s	0.20	1.84	0.14	2.27	2.52	-0.15	0.59	0.71	0.85	0.48	0.43	0.52
	$T=30$ s	-0.06	0.98	0.08	1.36	2.08	-0.19	0.41	0.52	0.60	0.54	0.49	0.67
	$T=60$ s	0.43	1.33	0.11	1.85	2.26	-0.24	0.44	0.51	0.72	0.56	0.56	0.79

Table C.20: Consistency summary statistics for the PSH-Road model applied to targets moving in rolling terrain (T2).

		$\mathbf{err}_{\mu_x}$ (m)	$\mathbf{err}_{\mu_y}$ (m)	$\mathbf{err}_{\mu_s}$ (m/s)	$\mathbf{err}_{\sigma_x}$ (m)	$\mathbf{err}_{\sigma_y}$ (m)	$\mathbf{err}_{\sigma_s}$ (m/s)	$\mathbf{rej}_{\mu_x}$ (%)	$\mathbf{rej}_{\mu_y}$ (%)	$\mathbf{rej}_{\mu_s}$ (%)	$\mathbf{rej}_{\sigma_x}$ (%)	$\mathbf{rej}_{\sigma_y}$ (%)	$\mathbf{rej}_{\sigma_s}$ (%)
PSH-Road/C1													
3 km	$T=15$ s	0.03	-0.09	0.07	-0.85	-0.95	-0.21	0.23	0.31	0.39	0.11	0.16	0.54
	$T=30$ s	0.01	-0.00	0.05	-0.89	-0.96	-0.27	0.11	0.23	0.34	0.14	0.17	0.83
	$T=60$ s	-0.01	-0.00	0.05	-0.78	-0.84	-0.40	0.22	0.30	0.52	0.08	0.15	0.87
5 km	$T=15$ s	0.08	-0.15	0.10	-1.30	-1.47	-0.22	0.61	0.60	0.79	0.27	0.33	0.55
	$T=30$ s	0.02	-0.00	0.06	-1.31	-1.45	-0.26	0.26	0.35	0.38	0.31	0.37	0.72
	$T=60$ s	0.09	-0.07	0.06	-1.31	-1.43	-0.41	0.29	0.44	0.50	0.23	0.32	0.85
7 km	$T=15$ s	0.18	-0.19	0.12	-1.23	-1.60	-0.22	0.75	0.69	0.88	0.34	0.39	0.55
	$T=30$ s	0.15	-0.11	0.07	-1.28	-1.70	-0.26	0.37	0.47	0.45	0.33	0.40	0.70
	$T=60$ s	0.12	-0.09	0.06	-1.64	-1.83	-0.43	0.28	0.53	0.52	0.30	0.37	0.84
PSH-Road/C2													
3 km	$T=15$ s	-0.03	0.08	0.05	-0.84	-0.87	-0.25	0.52	0.45	0.52	0.20	0.23	0.62
	$T=30$ s	0.03	0.12	0.05	-0.85	-0.92	-0.25	0.51	0.42	0.53	0.30	0.30	0.87
	$T=60$ s	-0.08	0.16	0.07	-0.83	-0.93	-0.23	0.40	0.35	0.84	0.27	0.32	0.95
5 km	$T=15$ s	-0.36	0.17	0.08	0.80	-0.30	-0.23	0.55	0.63	0.85	0.34	0.39	0.54
	$T=30$ s	0.05	0.12	0.05	-0.39	-0.78	-0.25	0.43	0.40	0.46	0.38	0.43	0.73
	$T=60$ s	0.04	0.18	0.07	-0.28	-0.67	-0.25	0.49	0.38	0.71	0.47	0.54	0.89
7 km	$T=15$ s	-1.37	-0.60	0.12	8.83	4.44	-0.19	0.69	0.47	0.93	0.49	0.56	0.45
	$T=30$ s	-0.53	-0.55	0.06	5.85	1.78	-0.23	0.34	0.35	0.48	0.41	0.48	0.58
	$T=60$ s	0.45	-0.13	0.06	0.06	-0.17	-0.26	0.51	0.35	0.64	0.57	0.57	0.82

Table C.21: Consistency summary statistics for the PSH-Road model applied to targets moving in mountain terrain (T3).

		$\mathbf{error_{xy}}$ (m)		$\mathbf{error_{s}}$ (m/s)		**divergence**		$\triangle\mathbf{error_{xy}}$ (%)		**area** (m^2)		$\triangle$**area** (%)	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
PSH-Road/C1													
3 km	$T=15$ s	6.31	0.24	0.189	0.025	4.92	0.14	-24.78	3.29	154.84	4.38	-31.41	2.41
	$T=30$ s	6.39	0.22	0.116	0.006	2.32	0.03	-24.27	2.66	161.54	4.13	-28.55	2.13
	$T=60$ s	6.74	0.21	0.092	0.007	1.21	0.02	-20.04	2.60	171.81	4.61	-24.08	2.32
5 km	$T=15$ s	7.98	0.73	0.227	0.053	5.13	0.24	-37.53	6.24	207.30	10.21	-44.75	2.97
	$T=30$ s	8.13	0.63	0.141	0.022	2.43	0.06	-37.26	4.97	222.35	13.26	-40.74	3.90
	$T=60$ s	8.76	0.64	0.107	0.015	1.26	0.02	-32.63	4.78	240.58	13.09	-35.96	3.88
7 km	$T=15$ s	9.25	1.33	0.249	0.070	5.25	0.32	-46.37	8.19	241.13	15.80	-54.11	3.20
	$T=30$ s	9.44	1.16	0.165	0.038	2.50	0.10	-46.20	6.72	264.58	23.86	-49.63	4.87
	$T=60$ s	10.29	1.08	0.123	0.019	1.28	0.03	-41.64	6.17	291.81	24.99	-44.50	5.13
PSH-Road/C2													
3 km	$T=15$ s	5.33	0.21	0.109	0.010	5.00	0.09	-36.20	2.69	112.79	6.11	-50.06	3.26
	$T=30$ s	5.62	0.10	0.112	0.003	2.58	0.01	-32.50	0.85	122.44	1.62	-45.93	0.57
	$T=60$ s	6.11	0.15	0.136	0.002	1.26	0.05	-26.33	2.23	142.62	1.20	-36.82	1.04
5 km	$T=15$ s	6.82	0.82	0.137	0.037	5.19	0.28	-46.30	6.94	155.01	16.81	-58.64	4.94
	$T=30$ s	6.99	0.13	0.109	0.009	2.77	0.18	-45.41	0.83	164.30	7.68	-56.30	2.28
	$T=60$ s	7.36	0.24	0.122	0.004	1.29	0.03	-42.28	1.84	183.03	4.59	-51.28	1.50
7 km	$T=15$ s	8.08	1.46	0.158	0.054	5.45	0.41	-52.81	8.87	181.87	24.04	-65.35	4.90
	$T=30$ s	7.98	0.73	0.118	0.019	2.68	0.11	-54.17	4.13	197.66	20.37	-62.38	4.18
	$T=60$ s	8.20	0.30	0.119	0.006	1.29	0.03	-52.65	1.97	210.44	9.15	-60.00	1.97

Table C.22: State estimation performance summary statistics for the PSH-Road model applied to targets moving in flat terrain (T1).

		$\mathbf{error_{xy}}$ (m)		$\mathbf{error_{s}}$ (m/s)		**divergence**		$\triangle\mathbf{error_{xy}}$ (%)		**area** (m^2)		$\triangle$**area** (%)	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
PSH-Road/C1													
3 km	$T=15$ s	6.60	0.20	0.228	0.018	5.21	0.08	-21.39	2.30	158.26	3.23	-30.11	1.52
	$T=30$ s	6.69	0.17	0.158	0.005	2.54	0.01	-20.80	1.82	165.53	3.75	-26.95	1.71
	$T=60$ s	7.14	0.23	0.134	0.010	1.30	0.06	-15.87	2.36	181.11	4.37	-20.19	1.95
5 km	$T=15$ s	8.53	0.49	0.265	0.026	5.53	0.09	-33.66	3.74	212.68	7.91	-43.54	2.16
	$T=30$ s	8.70	0.48	0.184	0.014	2.61	0.03	-33.02	3.38	229.32	11.35	-39.16	3.06
	$T=60$ s	9.69	0.69	0.163	0.012	1.39	0.10	-25.82	4.98	259.65	12.06	-31.19	3.21
7 km	$T=15$ s	10.34	0.70	0.285	0.030	5.92	0.09	-40.46	4.03	249.26	11.86	-52.74	2.28
	$T=30$ s	10.45	0.77	0.207	0.021	2.77	0.04	-40.48	4.20	274.13	18.92	-48.04	3.63
	$T=60$ s	11.89	1.22	0.182	0.020	1.41	0.08	-32.58	6.57	319.78	20.56	-39.46	3.90
PSH-Road/C2													
3 km	$T=15$ s	6.48	0.46	0.162	0.014	9.50	2.99	-22.06	5.66	120.10	4.10	-47.00	1.87
	$T=30$ s	6.45	0.29	0.163	0.003	3.78	0.29	-22.51	3.37	127.50	1.72	-43.72	0.82
	$T=60$ s	7.07	0.39	0.232	0.004	1.68	0.12	-15.26	4.47	148.14	1.72	-34.57	0.86
5 km	$T=15$ s	8.88	1.02	0.190	0.025	12.09	3.35	-30.15	8.06	161.52	8.11	-57.12	2.20
	$T=30$ s	8.40	0.57	0.162	0.003	4.37	0.62	-34.40	4.35	169.47	7.57	-55.03	2.03
	$T=60$ s	9.45	0.98	0.213	0.008	1.99	0.28	-25.85	7.65	191.44	5.69	-49.17	1.54
7 km	$T=15$ s	12.10	1.46	0.218	0.027	18.23	5.84	-29.70	8.20	186.90	11.45	-64.55	2.21
	$T=30$ s	10.69	1.20	0.174	0.005	5.65	0.94	-38.24	6.97	199.82	14.56	-62.11	2.80
	$T=60$ s	11.88	1.29	0.206	0.008	2.41	0.35	-31.35	7.17	219.70	9.20	-58.33	1.77

Table C.23: State estimation performance summary statistics for the PSH-Road model applied to targets moving in rolling terrain (T2).

		$\mathbf{error_{xy}}$ (m)		$\mathbf{error_{s}}$ (m/s)		**divergence**		$\triangle\mathbf{error_{xy}}$ (%)		**area** (m^2)		$\triangle$**area** (%)	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
PSH-Road/C1													
3 km	$T=15$ s	6.57	0.36	0.232	0.030	5.09	0.14	-22.00	3.52	156.28	7.39	-30.96	2.36
	$T=30$ s	6.62	0.32	0.155	0.012	2.44	0.03	-21.70	2.91	162.65	7.46	-28.15	2.38
	$T=60$ s	7.17	0.32	0.145	0.022	1.26	0.02	-15.55	2.86	179.28	7.66	-20.99	2.49
5 km	$T=15$ s	8.49	0.86	0.266	0.046	5.38	0.25	-34.14	6.49	210.84	15.23	-44.03	3.80
	$T=30$ s	8.74	0.87	0.186	0.025	2.58	0.09	-32.68	6.34	227.74	18.32	-39.56	4.67
	$T=60$ s	9.70	0.87	0.161	0.022	1.31	0.03	-25.80	5.99	258.60	20.12	-31.42	5.12
7 km	$T=15$ s	10.25	1.48	0.285	0.056	5.73	0.38	-41.06	8.50	246.73	23.20	-53.24	4.33
	$T=30$ s	10.54	1.40	0.210	0.034	2.73	0.12	-39.90	7.76	273.09	29.75	-48.25	5.62
	$T=60$ s	11.83	1.46	0.184	0.033	1.35	0.05	-33.25	7.80	321.32	34.49	-39.14	6.52
PSH-Road/C2													
3 km	$T=15$ s	5.88	0.19	0.154	0.007	5.70	0.21	-29.69	1.69	121.63	6.10	-46.32	2.14
	$T=30$ s	5.87	0.13	0.158	0.007	2.68	0.06	-29.48	0.98	129.14	2.45	-42.78	0.37
	$T=60$ s	6.45	0.23	0.187	0.003	1.39	0.03	-22.91	2.03	148.97	5.39	-34.08	1.41
5 km	$T=15$ s	8.32	1.24	0.186	0.026	11.40	6.06	-34.81	9.67	162.99	13.55	-56.75	3.46
	$T=30$ s	7.72	0.29	0.156	0.007	3.07	0.12	-39.85	2.21	174.65	8.32	-53.60	2.05
	$T=60$ s	8.41	0.96	0.176	0.004	1.55	0.19	-34.27	7.13	194.73	11.57	-48.22	2.76
7 km	$T=15$ s	14.20	4.35	0.220	0.042	30.34	19.66	-17.39	25.85	186.46	19.58	-64.67	3.69
	$T=30$ s	11.62	2.49	0.172	0.011	6.05	3.15	-32.73	14.90	206.38	16.30	-60.88	3.08
	$T=60$ s	10.07	1.55	0.174	0.009	1.69	0.20	-42.10	8.72	227.06	19.29	-56.92	3.57

Table C.24: State estimation performance summary statistics for the PSH-Road model applied to targets moving in mountain terrain (T3).

Appendix D

Inference Results

Prior	LL			Score			K			α				λ		
	MAP	Max Score	Truth	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Truth	Avg.	MAP	Max Score	Avg.
DP	-5233.8	-5409.5	-5406.9	0.02	0.99	0.09±0.20	49.0	1.0	36.1±13.50	14	0.0025	1e-06	12.4±8.57	8.1	15.9	9.7±2.09
DP (constr.)	-5293.9	-5409.5	-5406.9	0.06	0.99	0.14±0.26	22.0	1.0	25.4±11.32	6.86	0.0025	1e-06	4.81±3.87	11.3	15.9	11.0±1.83
DP (fixed α)	-5400.8	-5408.5	-5406.1	0.65	0.99	0.71±0.13	2.0	1.0	1.8±0.40	1e-06	1e-06	1e-06	1e-06±0.00	14.5	14.2	14.6±0.78
UP	-5219.0	-5408.8	-5406.0	0.03	0.99	0.04±0.09	48.0	1.0	41.2±11.89	5.64	0.0014	1e-06	5.76±3.62	7.2	15.9	8.5±1.90
UP (constr.)	-5227.6	-5408.9	-5406.1	0.03	0.99	0.06±0.09	39.0	1.0	29.8±8.16	3.87	0.00128	1e-06	1.93±1.33	9.5	15.9	10.1±1.49
UP (fixed α)	-5387.9	-5408.5	-5406.1	0.26	0.99	0.33±0.17	6.0	1.0	5.4±1.72	1e-06	1e-06	1e-06	1e-06±0.00	13.2	14.2	13.6±0.93
EP	-5202.6	-5409.2	-5406.0	0.02	0.99	0.04±0.10	58.0	1.0	53.1±15.38	2.55	0.00152	1e-06	2.53±1.85	6.1	15.9	7.0±2.34
EP (constr.)	-5229.0	-5409.2	-5406.0	0.02	0.99	0.03±0.09	51.0	1.0	50.1±12.10	2.94	0.00152	1e-06	1.11±0.79	7.3	15.9	7.4±1.95
EP (fixed α)	-5389.1	-5408.5	-5406.1	0.26	0.99	0.30±0.22	5.0	1.0	5.7±1.99	1e-06	1e-06	1e-06	1e-06±0.00	13.7	14.2	13.6±0.95
EP2	-5226.9	-5410.5	-5407.6	0.03	0.99	0.06±0.11	38.0	1.0	31.2±8.71	171	0.00457	1e-06	114±42.25	8.7	15.9	9.8±1.57
EP2 (constr.)	-5274.0	-5410.5	-5407.6	0.04	0.99	0.11±0.10	29.0	1.0	18.2±7.91	35.7	0.00457	1e-06	4.71±8.57	10.8	15.9	12.0±1.35
EP2 (fixed α)	-5403.7	-5408.5	-5406.1	0.32	0.99	0.32±0.21	4.0	1.0	5.1±1.66	1e-06	1e-06	1e-06	1e-06±0.00	13.9	14.2	13.7±0.92
HUP	-5150.1	-5409.5	-5407.1	0.02	0.99	0.07±0.13	49.0	1.0	38.1±15.44	N/A	N/A	N/A	N/A	7.5	14.2	9.1±2.51
HUP (constr.)	-5346.5	-5410.8	-5408.4	0.07	0.99	0.16±0.10	20.0	1.0	11.0±4.70	N/A	N/A	N/A	N/A	10.7	14.2	12.4±1.09
K	-4498.1	-5411.3	-5408.4	0.01	0.96	0.02±0.04	62.0	1.0	53.9±12.08	N/A	N/A	N/A	N/A	3.5	14.9	4.9±2.29
Prop.	-5429.8	-5441.1	-5438.7	0.34	0.99	0.36±0.04	4.0	1.0	3.8±0.44	N/A	N/A	N/A	N/A	14.6	14.2	14.6±0.78
Oh	-5220.1	-5436.1	-5420.2	0.05	0.99	0.10±0.08	24.0	1.0	16.8±5.03	N/A	N/A	N/A	N/A	10.5	15.1	12.1±1.15

Table D.1: Summary of results from the MCMC sampler on the one target data set.

Prior	**Target Score**					
	τ_1			Combined		
	MAP	Max Score	Avg.	MAP	Max Score	Avg.
DP	0.38	0.98	0.43±0.18	0.02	0.99	0.09±0.20
DP (constr.)	0.44	0.98	0.53±0.21	0.06	0.99	0.14±0.26
DP (fixed α)	0.98	0.98	0.96±0.03	0.65	0.99	0.71±0.13
UP	0.65	0.98	0.51±0.11	0.03	0.99	0.04±0.09
UP (constr.)	0.44	0.98	0.55±0.11	0.03	0.99	0.06±0.09
UP (fixed α)	0.88	0.98	0.89±0.05	0.26	0.99	0.33±0.17
EP	0.32	0.98	0.37±0.08	0.02	0.99	0.04±0.10
EP (constr.)	0.32	0.98	0.36±0.09	0.02	0.99	0.03±0.09
EP (fixed α)	0.61	0.98	0.65±0.10	0.26	0.99	0.30±0.22
EP2	0.44	0.98	0.46±0.08	0.03	0.99	0.06±0.11
EP2 (constr.)	0.38	0.98	0.60±0.13	0.04	0.99	0.11±0.10
EP2 (fixed α)	0.62	0.98	0.66±0.09	0.32	0.99	0.32±0.21
HUP	0.32	0.98	0.52±0.10	0.02	0.99	0.07±0.13
HUP (constr.)	0.53	0.98	0.63±0.08	0.07	0.99	0.16±0.10
K	0.32	0.93	0.50±0.10	0.01	0.96	0.02±0.04
Prop.	0.69	0.98	0.69±0.03	0.34	0.99	0.36±0.04
Oh	0.38	0.98	0.45±0.09	0.05	0.99	0.10±0.08

Table D.2: Target scores for the results from the MCMC sampler on the one target data set.

Prior	LL			Score			K			α				λ		
	MAP	Max Score	Truth	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Truth	Avg.	MAP	Max Score	Avg.
DP	-5354.4	-5408.7	-5406.9	0.16	0.99	0.29±0.21	11.0	1.0	7.3±2.86	3.01	0.0244	1e-06	2.77±2.49	12.9	15.1	13.4±0.94
DP (constr.)	-5343.7	-5408.7	-5406.9	0.21	0.99	0.56±0.31	8.0	1.0	3.9±2.72	1.97	0.0244	1e-06	0.394±0.72	13.4	15.1	14.2±0.94
DP (fixed α)	-5408.1	-5408.2	-5406.1	0.99	0.99	0.98±0.01	1.0	1.0	1.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.9	15.2	14.9±0.77
UP	-5314.7	-5408.5	-5406.0	0.13	0.99	0.21±0.18	10.0	1.0	8.2±2.69	0.501	0.0244	1e-06	0.914±0.80	12.6	15.1	13.2±0.92
UP (constr.)	-5374.9	-5408.5	-5406.1	0.32	0.99	0.45±0.12	5.0	1.0	3.5±0.82	0.139	0.0244	1e-06	0.0207±0.09	13.3	15.1	13.9±0.82
UP (fixed α)	-5400.5	-5408.2	-5406.1	0.65	0.99	0.73±0.14	2.0	1.0	1.7±0.43	1e-06	1e-06	1e-06	1e-06±0.00	14.4	15.2	14.6±0.79
EP	-5261.6	-5412.9	-5406.0	0.07	0.99	0.13±0.16	20.0	1.0	14.1±4.50	0.826	0.0244	1e-06	0.783±0.66	12.4	15.1	13.0±0.91
EP (constr.)	-5276.9	-5412.9	-5406.0	0.08	0.99	0.28±0.22	17.0	1.0	8.3±5.52	0.994	0.0244	1e-06	0.296±0.46	12.9	15.1	13.7±0.93
EP (fixed α)	-5408.1	-5408.2	-5406.1	0.99	0.99	0.98±0.01	1.0	1.0	1.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.9	15.2	14.9±0.77
EP2	-5265.3	-5410.1	-5407.6	0.07	0.99	0.17±0.17	19.0	1.0	11.6±5.75	47.6	0.0244	1e-06	31.1±24.55	12.1	15.1	12.9±0.99
EP2 (constr.)	-5371.1	-5410.1	-5407.6	0.39	0.99	0.40±0.10	4.0	1.0	3.9±0.55	0.0289	0.0244	1e-06	0.0183±0.01	14.9	15.1	13.7±0.77
EP2 (fixed α)	-5408.1	-5408.2	-5406.1	0.99	0.99	0.98±0.01	1.0	1.0	1.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.9	15.2	14.9±0.77
HUP	-5332.5	-5409.2	-5407.1	0.15	0.99	0.37±0.22	11.0	1.0	5.8±3.37	N/A	N/A	N/A	N/A	13.0	15.2	13.8±0.95
HUP (constr.)	-5371.7	-5410.5	-5408.4	0.28	0.99	0.36±0.17	5.0	1.0	4.2±1.01	N/A	N/A	N/A	N/A	13.3	15.2	13.7±0.86
K	-4557.0	-5411.4	-5408.4	0.01	0.96	0.02±0.03	55.0	1.0	46.5±9.02	N/A	N/A	N/A	N/A	4.1	15.1	5.4±1.70
Prop.	-5440.1	-5440.8	-5438.7	0.65	0.99	0.52±0.17	2.0	1.0	3.5±2.25	N/A	N/A	N/A	N/A	14.5	15.2	14.1±1.03
Oh	-5284.2	-5436.1	-5420.2	0.06	0.99	0.13±0.12	22.0	1.0	13.4±5.21	N/A	N/A	N/A	N/A	11.6	15.1	11.9±1.27

Table D.3: Summary of results from the MCMC sampler on the one target data set. The individual moves were allowed to propose remaining in the current state.

Prior	**Target Score**					
	τ_1			Combined		
	MAP	Max Score	Avg.	MAP	Max Score	Avg.
DP	0.98	0.98	0.96±0.02	0.16	0.99	0.29±0.21
DP (constr.)	0.98	0.98	0.96±0.02	0.21	0.99	0.56±0.31
DP (fixed α)	0.98	0.98	0.96±0.02	0.99	0.99	0.98±0.01
UP	0.54	0.98	0.61±0.14	0.13	0.99	0.21±0.18
UP (constr.)	0.98	0.98	0.96±0.02	0.32	0.99	0.45±0.12
UP (fixed α)	0.98	0.98	0.96±0.02	0.65	0.99	0.73±0.14
EP	0.44	0.98	0.45±0.14	0.07	0.99	0.13±0.16
EP (constr.)	0.44	0.98	0.58±0.18	0.08	0.99	0.28±0.22
EP (fixed α)	0.98	0.98	0.96±0.02	0.99	0.99	0.98±0.01
EP2	0.32	0.98	0.48±0.16	0.07	0.99	0.17±0.17
EP2 (constr.)	0.98	0.98	0.96±0.02	0.39	0.99	0.40±0.10
EP2 (fixed α)	0.98	0.98	0.96±0.02	0.99	0.99	0.98±0.01
HUP	0.93	0.98	0.96±0.02	0.15	0.99	0.37±0.22
HUP (constr.)	0.73	0.98	0.76±0.08	0.28	0.99	0.36±0.17
K	0.38	0.93	0.40±0.03	0.01	0.96	0.02±0.03
Prop.	0.98	0.98	0.95±0.03	0.65	0.99	0.52±0.17
Oh	0.37	0.98	0.48±0.13	0.06	0.99	0.13±0.12

Table D.4: Target scores for the results from the MCMC sampler on the one target data set. The individual moves were allowed to propose remaining in the current state.

Prior	LL			Score			K			α				λ		
	MAP	Max Score	Truth	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Truth	Avg.	MAP	Max Score	Avg.
DP	-5352.8	-5356.1	-5406.2	0.24	0.24	0.24±0.00	7.0	7.0	7.0±0.00	1.01	1.34	1e-06	1.78±0.64	12.9	14.5	13.0±0.38
DP (constr.)	-5352.1	-5355.8	-5406.2	0.24	0.24	0.24±0.00	7.0	7.0	7.0±0.00	0.653	0.696	1e-06	0.711±0.16	13.0	12.4	13.2±0.39
DP (fixed α)	-5395.8	-5395.8	-5406.1	0.64	0.64	0.64±0.00	2.0	2.0	2.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.2	14.2	14.2±0.00
UP	-5334.8	-5338.9	-5406.0	0.20	0.20	0.20±0.00	7.0	7.0	7.0±0.00	0.246	0.422	1e-06	0.919±0.38	13.0	13.5	13.1±0.30
UP (constr.)	-5325.7	-5328.4	-5406.1	0.21	0.21	0.21±0.00	7.0	7.0	7.0±0.00	0.275	0.23	1e-06	0.292±0.04	14.5	13.0	13.0±0.20
UP (fixed α)	-5389.1	-5392.6	-5406.1	0.52	0.54	0.53±0.01	2.0	2.0	2.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.5	13.4	13.9±0.65
EP	-5344.8	-5345.0	-5406.1	0.19	0.19	0.19±0.00	8.0	8.0	8.0±0.00	0.149	0.134	1e-06	0.0811±0.04	13.4	13.2	13.2±0.24
EP (constr.)	-5344.9	-5348.4	-5406.1	0.16	0.16	0.16±0.00	8.0	8.0	8.0±0.00	0.0486	0.0731	1e-06	0.073±0.00	13.0	11.4	11.4±0.02
EP (fixed α)	-5394.4	-5396.3	-5406.1	0.53	0.54	0.53±0.01	2.0	2.0	2.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	13.9	13.5	14.0±0.45
EP2	-5348.2	-5352.2	-5406.1	0.18	0.18	0.18±0.00	8.0	8.0	8.0±0.00	4.5	10.9	1e-06	3.91±1.81	12.5	12.4	12.5±0.47
EP2 (constr.)	-5337.2	-5339.5	-5406.1	0.18	0.18	0.18±0.00	8.0	8.0	8.0±0.00	0.129	0.199	1e-06	0.129±0.00	13.2	12.7	13.2±0.06
EP2 (fixed α)	-5391.9	-5398.1	-5406.1	0.54	0.54	0.54±0.00	2.0	2.0	2.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.5	13.3	14.3±0.31
HUP	-5337.5	-5341.0	-5407.1	0.19	0.19	0.19±0.00	8.0	8.0	8.0±0.00	N/A	N/A	N/A	N/A	13.0	13.6	12.8±0.68
HUP (constr.)	-5306.6	-5306.6	-5406.1	0.18	0.18	0.18±0.00	8.0	8.0	8.0±0.00	N/A	N/A	N/A	N/A	13.8	13.8	13.8±0.00
K	-5148.8	-5148.8	-5406.1	0.12	0.12	0.12±0.00	14.0	14.0	14.0±0.00	N/A	N/A	N/A	N/A	10.1	10.1	10.1±0.00
Prop.	-5412.4	-5416.8	-5438.7	0.18	0.18	0.18±0.00	8.0	8.0	8.0±0.00	N/A	N/A	N/A	N/A	12.1	13.9	12.1±0.07
Oh	-5334.0	-5336.1	-5420.2	0.19	0.19	0.19±0.00	8.0	8.0	8.0±0.00	N/A	N/A	N/A	N/A	13.3	12.8	13.5±0.30

Table D.5: Summary of results from the PMC sampler on the one target data set.

Prior	**Target Score**					
	τ_1			Combined		
	MAP	Max Score	Avg.	MAP	Max Score	Avg.
DP	0.98	0.98	0.98±0.01	0.24	0.24	0.24±0.00
DP (constr.)	0.98	0.98	0.98±0.00	0.24	0.24	0.24±0.00
DP (fixed α)	0.93	0.93	0.93±0.00	0.64	0.64	0.64±0.00
UP	0.62	0.62	0.63±0.00	0.20	0.20	0.20±0.00
UP (constr.)	0.73	0.73	0.73±0.00	0.21	0.21	0.21±0.00
UP (fixed α)	0.58	0.61	0.59±0.01	0.52	0.54	0.53±0.01
EP	0.73	0.73	0.73±0.00	0.19	0.19	0.19±0.00
EP (constr.)	0.50	0.50	0.50±0.00	0.16	0.16	0.16±0.00
EP (fixed α)	0.59	0.61	0.60±0.01	0.53	0.54	0.53±0.01
EP2	0.67	0.67	0.67±0.00	0.18	0.18	0.18±0.00
EP2 (constr.)	0.67	0.67	0.67±0.00	0.18	0.18	0.18±0.00
EP2 (fixed α)	0.63	0.63	0.63±0.01	0.54	0.54	0.54±0.00
HUP	0.69	0.69	0.69±0.00	0.19	0.19	0.19±0.00
HUP (constr.)	0.67	0.67	0.67±0.00	0.18	0.18	0.18±0.00
K	0.93	0.93	0.93±0.00	0.12	0.12	0.12±0.00
Prop.	0.69	0.69	0.69±0.00	0.18	0.18	0.18±0.00
Oh	0.73	0.73	0.73±0.00	0.19	0.19	0.19±0.00

Table D.6: Target scores for the results from the PMC sampler on the one target data set.

Prior	LL			Score			K			α				λ		
	MAP	Max Score	Truth	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Truth	Avg.	MAP	Max Score	Avg.
DP	-5405.0	-5411.3	-5406.2	0.65	0.99	0.64±0.01	2.0	1.0	2.0±0.02	0.206	0.527	1e-06	0.625±0.22	14.4	15.9	14.7±0.53
DP (constr.)	-5388.7	-5393.5	-5406.2	0.32	0.32	0.32±0.00	5.0	5.0	5.0±0.00	0.741	0.285	1e-06	0.716±0.10	13.6	14.6	13.6±0.23
DP (fixed α)	-5408.1	-5409.7	-5406.1	0.99	0.99	0.99±0.01	1.0	1.0	1.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.9	13.5	15.0±0.77
UP	-5350.2	-5350.2	-5406.0	0.24	0.24	0.21±0.00	7.0	7.0	8.0±0.00	0.168	0.168	1e-06	0.127±0.05	13.9	13.9	12.2±0.19
UP (constr.)	-5373.8	-5376.2	-5406.1	0.27	0.27	0.24±0.00	6.0	6.0	7.0±0.02	0.246	0.533	1e-06	0.168±0.00	13.4	12.4	12.7±0.02
UP (fixed α)	-5408.1	-5408.9	-5406.1	0.99	0.99	0.97±0.07	1.0	1.0	1.0±0.20	1e-06	1e-06	1e-06	1e-06±0.00	14.9	15.9	14.9±0.80
EP	-5365.0	-5377.7	-5406.1	0.28	0.29	0.29±0.00	5.0	5.0	5.0±0.00	0.0901	0.0398	1e-06	0.135±0.03	13.7	14.5	14.6±0.51
EP (constr.)	-5346.3	-5353.3	-5406.1	0.18	0.20	0.18±0.00	8.0	7.0	8.0±0.00	0.236	0.214	1e-06	0.473±0.07	13.1	13.9	15.1±0.45
EP (fixed α)	-5408.1	-5408.1	-5406.1	0.99	0.99	0.99±0.01	1.0	1.0	1.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.9	14.8	14.9±0.74
EP2	-5355.0	-5365.0	-5406.1	0.20	0.20	0.20±0.00	7.0	7.0	7.0±0.00	5.95	14.3	1e-06	6±1.49	13.5	13.7	13.5±0.11
EP2 (constr.)	-5393.6	-5397.6	-5406.1	0.58	0.58	0.58±0.00	2.0	2.0	2.0±0.00	0.0187	0.0187	1e-06	0.0155±0.00	14.6	15.4	14.1±0.40
EP2 (fixed α)	-5408.1	-5408.1	-5406.1	0.99	0.99	0.98±0.01	1.0	1.0	1.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.9	14.8	14.8±0.73
HUP	-5352.8	-5352.8	-5407.1	0.23	0.23	0.23±0.00	7.0	7.0	7.0±0.00	N/A	N/A	N/A	N/A	14.1	14.1	14.0±0.34
HUP (constr.)	-5393.2	-5393.8	-5406.1	0.38	0.38	0.38±0.00	4.0	4.0	4.0±0.00	N/A	N/A	N/A	N/A	13.4	12.7	13.5±0.75
K	-5180.7	-5188.1	-5406.1	0.10	0.12	0.10±0.00	14.0	14.0	14.0±0.00	N/A	N/A	N/A	N/A	10.2	10.4	9.8±0.38
Prop.	-5417.1	-5417.3	-5438.7	0.32	0.39	0.27±0.01	5.0	4.0	6.0±0.12	N/A	N/A	N/A	N/A	13.2	14.0	13.6±0.31
Oh	-5339.7	-5341.3	-5420.2	0.18	0.18	0.18±0.00	8.0	8.0	8.0±0.00	N/A	N/A	N/A	N/A	12.6	12.7	13.0±0.40

Table D.7: Summary of results from the PMC sampler on the one target data set. The individual moves were allowed to propose remaining in the current state.

Prior	**Target Score**					
	τ_1			Combined		
	MAP	Max Score	Avg.	MAP	Max Score	Avg.
DP	0.98	0.98	0.94±0.02	0.65	0.99	0.64±0.01
DP (constr.)	0.98	0.98	0.98±0.00	0.32	0.32	0.32±0.00
DP (fixed α)	0.98	0.98	0.97±0.01	0.99	0.99	0.99±0.01
UP	0.98	0.98	0.98±0.00	0.24	0.24	0.21±0.00
UP (constr.)	0.98	0.98	0.98±0.00	0.27	0.27	0.24±0.00
UP (fixed α)	0.98	0.98	0.97±0.01	0.99	0.99	0.97±0.07
EP	0.73	0.73	0.73±0.00	0.28	0.29	0.29±0.00
EP (constr.)	0.65	0.65	0.65±0.00	0.18	0.20	0.18±0.00
EP (fixed α)	0.98	0.98	0.97±0.01	0.99	0.99	0.99±0.01
EP2	0.65	0.65	0.65±0.00	0.20	0.20	0.20±0.00
EP2 (constr.)	0.74	0.76	0.74±0.00	0.58	0.58	0.58±0.00
EP2 (fixed α)	0.98	0.98	0.97±0.01	0.99	0.99	0.98±0.01
HUP	0.93	0.93	0.93±0.00	0.23	0.23	0.23±0.00
HUP (constr.)	0.93	0.93	0.93±0.00	0.38	0.38	0.38±0.00
K	0.65	0.93	0.65±0.01	0.10	0.12	0.10±0.00
Prop.	0.98	0.98	0.98±0.00	0.27	0.32	0.24±0.01
Oh	0.69	0.69	0.69±0.00	0.18	0.18	0.18±0.00

Table D.8: Target scores for the results from the PMC sampler on the one target data set. The individual moves were allowed to propose remaining in the current state.

Prior	LL			Score			K			α				λ		
	MAP	Max Score	Truth	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Truth	Avg.	MAP	Max Score	Avg.
DP	-5399.1	-5401.9	-5406.2	0.64	0.66	0.64±0.00	2.0	2.0	2.0±0.00	1.26	0.305	1e-06	0.936±0.18	14.3	14.9	15.3±0.40
DP (constr.)	-5409.4	-5410.4	-5406.2	0.65	0.96	0.65±0.00	2.0	1.0	2.0±0.00	0.0092	0.0062	1e-06	0.011±0.00	14.5	13.6	14.2±0.06
DP (fixed α)	-5408.1	-5408.2	-5406.1	0.99	0.99	0.98±0.01	1.0	1.0	1.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.9	15.1	14.9±0.78
UP	-5407.5	-5407.5	-5406.0	0.39	0.39	0.38±0.00	4.0	4.0	4.0±0.00	0.0994	0.0994	1e-06	0.869±0.03	13.9	13.9	15.2±0.05
UP (constr.)	-5397.3	-5397.3	-5406.1	0.65	0.65	0.65±0.00	2.0	2.0	2.0±0.00	0.004	0.004	1e-06	0.004±0.00	14.9	14.9	14.9±0.00
UP (fixed α)	-5408.1	-5408.9	-5406.1	0.99	0.99	0.66±0.06	1.0	1.0	2.0±0.17	1e-06	1e-06	1e-06	1e-06±0.00	14.8	15.9	14.3±0.36
EP	-5408.2	-5408.2	-5406.1	0.99	0.99	0.78±0.16	1.0	1.0	1.6±0.49	0.000188	0.000629	1e-06	0.0113±0.01	15.2	14.8	14.7±0.68
EP (constr.)	-5405.2	-5405.2	-5406.1	0.65	0.65	0.64±0.02	2.0	2.0	2.0±0.09	0.00205	0.00205	1e-06	0.0108±0.00	14.1	14.1	14.0±0.07
EP (fixed α)	-5408.1	-5408.1	-5406.1	0.99	0.99	0.98±0.01	1.0	1.0	1.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.9	14.7	14.9±0.79
EP2	-5398.7	-5398.7	-5406.1	0.39	0.39	0.39±0.00	4.0	4.0	4.0±0.00	2.49	2.49	1e-06	2.49±0.00	13.9	13.9	13.9±0.00
EP2 (constr.)	-5406.5	-5410.0	-5406.1	0.64	0.99	0.64±0.01	2.0	1.0	2.0±0.02	0.0181	0.0141	1e-06	0.018±0.00	14.2	13.9	14.2±0.13
EP2 (fixed α)	-5408.1	-5409.6	-5406.1	0.99	0.99	0.98±0.01	1.0	1.0	1.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.9	16.2	15.0±0.82
HUP	-5409.6	-5415.4	-5407.1	0.49	0.49	0.49±0.00	3.0	3.0	3.0±0.00	N/A	N/A	N/A	N/A	14.8	14.7	14.7±0.36
HUP (constr.)	-5408.1	-5408.2	-5406.1	0.99	0.99	0.98±0.01	1.0	1.0	1.0±0.00	N/A	N/A	N/A	N/A	14.9	14.7	14.9±0.80
K	-5306.9	-5306.9	-5406.1	0.26	0.26	0.26±0.00	6.0	6.0	6.0±0.00	N/A	N/A	N/A	N/A	13.2	13.2	13.2±0.00
Prop.	-5440.7	-5441.2	-5438.7	0.99	0.99	0.98±0.01	1.0	1.0	1.0±0.00	N/A	N/A	N/A	N/A	14.9	14.1	14.9±0.77
Oh	-5419.2	-5420.6	-5420.2	0.99	0.99	0.98±0.01	1.0	1.0	1.0±0.00	N/A	N/A	N/A	N/A	15.1	14.4	15.0±0.79

Table D.9: Summary of results from the PMC sampler, using the k-add mixture proposal, on the one target data set.

Prior	**Target Score**					
	τ_1			Combined		
	MAP	Max Score	Avg.	MAP	Max Score	Avg.
DP	0.93	0.98	0.93±0.01	0.64	0.66	0.64±0.00
DP (constr.)	0.98	0.93	0.97±0.01	0.65	0.96	0.65±0.00
DP (fixed α)	0.98	0.98	0.97±0.02	0.99	0.99	0.98±0.01
UP	0.98	0.98	0.93±0.00	0.39	0.39	0.38±0.00
UP (constr.)	0.98	0.98	0.98±0.00	0.65	0.65	0.65±0.00
UP (fixed α)	0.98	0.98	0.97±0.01	0.99	0.99	0.66±0.06
EP	0.98	0.98	0.97±0.02	0.99	0.99	0.78±0.16
EP (constr.)	0.98	0.98	0.94±0.02	0.65	0.65	0.64±0.02
EP (fixed α)	0.98	0.98	0.97±0.02	0.99	0.99	0.98±0.01
EP2	0.98	0.98	0.98±0.00	0.39	0.39	0.39±0.00
EP2 (constr.)	0.93	0.98	0.93±0.01	0.64	0.99	0.64±0.01
EP2 (fixed α)	0.98	0.98	0.97±0.02	0.99	0.99	0.98±0.01
HUP	0.98	0.98	0.97±0.01	0.49	0.49	0.49±0.00
HUP (constr.)	0.98	0.98	0.97±0.02	0.99	0.99	0.98±0.01
K	0.93	0.93	0.93±0.00	0.26	0.26	0.26±0.00
Prop.	0.98	0.98	0.97±0.02	0.99	0.99	0.98±0.01
Oh	0.98	0.98	0.97±0.02	0.99	0.99	0.98±0.01

Table D.10: Target scores for the results from the PMC sampler, using the k-add mixture proposal, on the one target data set.

Prior	LL			Score			K			α				λ		
	MAP	Max Score	Truth	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Truth	Avg.	MAP	Max Score	Avg.
DP	-15265.1	-15331.3	-15341.3	0.08	0.82	0.10±0.10	57.0	5.0	55.1±18.02	15	1.05	0.599	20.9±15.43	11.5	14.9	11.7±1.17
DP (constr.)	-15240.3	-15340.6	-15345.8	0.10	0.82	0.11±0.11	44.0	5.0	54.9±16.47	14.9	0.0279	0.0959	16±12.19	12.0	14.9	11.7±1.09
DP (fixed α)	-15366.1	-15366.2	-15389.2	0.87	0.87	0.77±0.08	5.0	5.0	5.2±0.75	1e-06	1e-06	1e-06	1e-06±0.00	14.4	14.2	14.5±0.49
UP	-15104.8	-15355.5	-15304.0	0.03	0.83	0.09±0.12	107.0	5.0	76.2±31.81	14.6	1.45	0.0773	11.1±8.18	9.3	15.0	10.9±1.61
UP (constr.)	-15163.0	-15392.8	-15306.3	0.04	0.77	0.10±0.12	92.0	5.0	65.9±29.70	8.9	0.00847	0.0195	6.5±5.33	10.0	14.4	11.3±1.56
UP (fixed α)	-15324.7	-15377.3	-15344.1	0.48	0.76	0.46±0.08	9.0	5.0	9.5±1.82	1e-06	1e-06	1e-06	1e-06±0.00	13.9	13.8	13.9±0.52
EP	-15200.9	-15416.2	-15389.2	0.04	0.80	0.07±0.13	80.0	4.0	96.5±34.24	3.36	0.00349	0.0154	4.68±3.78	11.1	13.9	10.0±1.94
EP (constr.)	-15271.5	-15466.9	-15388.7	0.05	0.85	0.13±0.21	64.0	5.0	76.8±39.58	2.31	0.164	0.014	2.35±2.17	12.3	15.0	11.0±2.22
EP (fixed α)	-15409.7	-15429.5	-15419.7	0.63	0.76	0.63±0.06	6.0	6.0	6.0±0.66	1e-06	1e-06	1e-06	1e-06±0.00	14.4	14.8	14.4±0.51
EP2	-15178.8	-15389.5	-15347.0	0.03	0.80	0.09±0.14	93.0	5.0	73.4±28.18	391	5.16	1.9	304±120.83	9.9	14.4	11.0±1.57
EP2 (constr.)	-15207.6	-15393.0	-15340.7	0.03	0.82	0.12±0.19	102.0	5.0	73.9±35.27	436	0.322	0.197	259±148.49	9.7	14.6	10.8±1.85
EP2 (fixed α)	-15353.6	-15384.4	-15382.1	0.74	0.79	0.74±0.03	6.0	6.0	5.9±0.54	1e-06	1e-06	1e-06	1e-06±0.00	14.2	14.0	14.3±0.50
HUP	-14930.5	-15379.3	-15337.1	0.03	0.80	0.06±0.09	110.0	5.0	94.5±26.89	N/A	N/A	N/A	N/A	8.7	14.5	9.6±1.64
HUP (constr.)	-15155.3	-15349.6	-15357.9	0.65	0.78	0.59±0.07	4.0	5.0	6.1±1.86	N/A	N/A	N/A	N/A	15.8	14.4	14.3±0.50
K	-13518.8	-15320.1	-15157.2	0.03	0.62	0.05±0.07	127.0	4.0	102.2±28.49	N/A	N/A	N/A	N/A	5.3	15.3	6.9±2.15
Prop.	-15236.2	-15499.3	-15388.7	0.09	0.69	0.12±0.07	53.0	5.0	44.1±12.61	N/A	N/A	N/A	N/A	11.6	15.4	12.1±0.96
Oh	-15216.6	-15420.3	-15418.1	0.13	0.57	0.16±0.06	25.0	3.0	21.6±4.14	N/A	N/A	N/A	N/A	14.5	15.1	14.5±0.50

Table D.11: Summary of results from the MCMC sampler on the five target data set.

Prior	Target Score																	
	τ_1			τ_2			τ_3			τ_4			τ_5			Combined		
	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.
DP	0.65	0.65	0.66±0.06	0.97	0.99	0.94±0.11	0.26	0.53	0.32±0.10	0.92	0.95	0.79±0.18	0.46	0.62	0.55±0.13	0.08	0.82	0.10±0.10
DP (constr.)	0.65	0.65	0.66±0.08	1.00	0.99	0.98±0.03	0.39	0.53	0.37±0.11	0.95	0.95	0.78±0.18	0.62	0.62	0.58±0.20	0.10	0.82	0.11±0.11
DP (fixed α)	0.72	0.72	0.66±0.07	0.99	0.99	0.98±0.02	0.86	0.86	0.67±0.15	0.92	0.92	0.92±0.04	0.74	0.74	0.60±0.28	0.87	0.87	0.77±0.08
UP	0.50	0.71	0.61±0.16	0.31	0.99	0.59±0.27	0.41	0.81	0.42±0.17	0.36	0.77	0.54±0.20	0.62	0.67	0.61±0.10	0.03	0.83	0.09±0.12
UP (constr.)	0.36	0.64	0.60±0.17	0.63	0.99	0.70±0.25	0.41	0.45	0.41±0.13	0.76	0.78	0.69±0.16	0.67	0.71	0.62±0.14	0.04	0.77	0.10±0.12
UP (fixed α)	0.64	0.53	0.61±0.06	0.99	0.99	0.99±0.02	0.49	0.48	0.48±0.04	0.89	0.92	0.90±0.04	0.76	0.57	0.69±0.15	0.48	0.76	0.46±0.08
EP	0.43	0.97	0.46±0.14	0.28	1.00	0.29±0.14	0.36	0.90	0.36±0.10	0.50	0.95	0.51±0.14	0.67	0.00	0.56±0.14	0.04	0.80	0.07±0.13
EP (constr.)	0.43	0.94	0.57±0.16	0.25	0.97	0.36±0.22	0.36	0.53	0.37±0.13	0.43	0.97	0.55±0.16	0.67	0.67	0.58±0.18	0.05	0.85	0.13±0.21
EP (fixed α)	1.00	1.00	0.99±0.06	0.78	0.99	0.82±0.08	0.48	0.57	0.51±0.06	0.92	0.92	0.87±0.06	0.22	0.78	0.17±0.16	0.63	0.76	0.63±0.06
EP2	0.50	0.65	0.68±0.16	0.28	1.00	0.41±0.19	0.31	0.43	0.31±0.07	0.43	0.89	0.55±0.13	0.62	0.78	0.60±0.14	0.03	0.80	0.09±0.14
EP2 (constr.)	0.56	0.64	0.62±0.12	0.25	0.99	0.44±0.24	0.26	0.50	0.36±0.14	0.36	0.90	0.60±0.20	0.57	0.72	0.47±0.15	0.03	0.82	0.12±0.19
EP2 (fixed α)	1.00	0.97	0.98±0.08	0.99	0.99	0.99±0.02	0.52	0.82	0.52±0.07	0.83	0.92	0.89±0.06	0.86	0.86	0.75±0.28	0.74	0.79	0.74±0.03
HUP	0.50	0.65	0.53±0.09	0.99	0.99	0.99±0.02	0.41	0.48	0.41±0.14	0.40	0.92	0.57±0.13	0.57	0.67	0.57±0.05	0.03	0.80	0.06±0.09
HUP (constr.)	0.64	0.64	0.61±0.07	1.00	0.99	0.99±0.02	0.48	0.49	0.48±0.04	0.75	0.73	0.79±0.05	0.00	0.76	0.44±0.41	0.65	0.78	0.59±0.07
K	0.84	0.68	0.63±0.07	0.49	0.99	0.50±0.12	0.31	0.18	0.31±0.02	0.59	0.73	0.61±0.12	0.71	0.00	0.70±0.10	0.03	0.62	0.05±0.07
Prop.	0.85	0.74	0.85±0.04	0.88	0.87	0.86±0.03	0.50	0.26	0.42±0.04	0.79	0.92	0.92±0.06	0.62	0.00	0.62±0.05	0.09	0.69	0.12±0.07
Oh	0.43	0.64	0.51±0.13	0.33	0.97	0.37±0.11	0.31	0.00	0.36±0.05	0.40	0.78	0.44±0.08	0.71	0.00	0.55±0.30	0.13	0.57	0.16±0.06

Table D.12: Target scores for the results from the MCMC sampler on the five target data set.

Prior	LL			Score			K			α				λ		
	MAP	Max Score	Truth	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Truth	Avg.	MAP	Max Score	Avg.
DP	-15325.3	-15347.0	-15341.3	0.69	0.71	0.68±0.02	4.0	4.0	4.0±0.19	0.581	1.78	0.599	2.83±3.66	14.4	15.7	14.6±0.50
DP (constr.)	-15321.5	-15359.5	-15345.8	0.68	0.70	0.68±0.01	4.0	4.0	4.0±0.14	0.0685	0.0481	0.0959	0.0361±0.10	14.7	15.5	14.6±0.49
DP (fixed α)	-15379.9	-15389.5	-15389.2	0.58	0.60	0.58±0.01	3.0	3.0	3.0±0.08	1e-06	1e-06	1e-06	1e-06±0.00	14.9	14.2	14.9±0.49
UP	-15301.2	-15367.6	-15304.0	0.66	0.90	0.70±0.06	4.0	5.0	3.9±0.31	0.0684	1.2	0.0773	0.496±0.84	14.6	15.6	14.6±0.50
UP (constr.)	-15295.6	-15316.3	-15306.3	0.68	0.69	0.66±0.02	4.0	4.0	4.0±0.18	0.0264	0.0267	0.0195	0.0141±0.02	14.7	14.4	14.6±0.49
UP (fixed α)	-15297.5	-15328.4	-15344.1	0.64	0.69	0.65±0.02	4.0	4.0	4.0±0.14	1e-06	1e-06	1e-06	1e-06±0.00	14.4	13.9	14.4±0.49
EP	-15389.8	-15424.8	-15389.2	0.70	0.80	0.63±0.09	4.0	4.0	4.1±0.87	0.0174	0.018	0.0154	0.284±0.42	14.9	15.2	14.9±0.53
EP (constr.)	-15369.6	-15438.2	-15388.7	0.66	0.76	0.68±0.05	4.0	4.0	3.9±0.35	0.00409	0.00978	0.014	0.017±0.09	14.3	14.6	14.6±0.51
EP (fixed α)	-15412.3	-15414.6	-15419.7	0.60	0.60	0.58±0.02	3.0	3.0	3.0±0.08	1e-06	1e-06	1e-06	1e-06±0.00	14.9	14.3	14.9±0.49
EP2	-15320.5	-15471.4	-15347.0	0.66	0.81	0.64±0.04	4.0	5.0	3.9±0.50	0.303	44	1.9	8.24±14.10	15.1	13.7	14.6±0.52
EP2 (constr.)	-15308.9	-15361.5	-15340.7	0.67	0.68	0.67±0.01	4.0	4.0	4.0±0.11	0.303	0.417	0.197	0.209±0.12	14.2	14.3	14.5±0.49
EP2 (fixed α)	-15331.8	-15357.2	-15382.1	0.67	0.69	0.67±0.02	4.0	4.0	4.0±0.14	1e-06	1e-06	1e-06	1e-06±0.00	14.5	13.9	14.5±0.48
HUP	-15314.4	-15320.8	-15337.1	0.69	0.69	0.69±0.02	4.0	4.0	4.0±0.12	N/A	N/A	N/A	N/A	14.6	14.6	14.6±0.48
HUP (constr.)	-15087.9	-15374.6	-15357.9	0.64	0.78	0.65±0.04	4.0	5.0	4.0±0.28	N/A	N/A	N/A	N/A	14.4	14.0	14.4±0.50
K	-13462.9	-15302.1	-15157.2	0.03	0.55	0.05±0.05	122.0	6.0	99.1±27.33	N/A	N/A	N/A	N/A	6.4	15.2	7.5±1.82
Prop.	-15297.8	-15416.3	-15388.7	0.10	0.71	0.13±0.06	45.0	6.0	39.5±9.76	N/A	N/A	N/A	N/A	11.8	15.1	12.2±0.80
Oh	-15218.1	-15376.3	-15418.1	0.14	0.67	0.24±0.09	25.0	5.0	17.4±4.40	N/A	N/A	N/A	N/A	14.3	14.4	14.3±0.50

Table D.13: Summary of results from the MCMC sampler on the five target data set. The individual moves were allowed to propose remaining in the current state.

Prior	Target Score																	
	τ_1			τ_2			τ_3			τ_4			τ_5			Combined		
	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.
DP	0.67	0.69	0.66±0.04	0.99	1.00	0.99±0.01	0.53	0.53	0.50±0.06	0.92	0.97	0.86±0.09	0.00	0.00	0.00±0.00	0.69	0.71	0.68±0.02
DP (constr.)	0.65	0.65	0.65±0.01	0.99	1.00	0.99±0.01	0.52	0.53	0.51±0.06	0.87	0.95	0.91±0.05	0.00	0.00	0.00±0.00	0.68	0.70	0.68±0.01
DP (fixed α)	0.64	0.64	0.64±0.03	0.99	0.99	0.99±0.01	0.00	0.00	0.00±0.00	0.83	0.92	0.84±0.05	0.00	0.00	0.00±0.00	0.58	0.60	0.58±0.01
UP	0.64	1.00	0.65±0.13	0.97	0.97	0.98±0.01	0.51	0.93	0.65±0.27	0.82	0.77	0.86±0.10	0.00	0.74	0.01±0.09	0.66	0.90	0.70±0.06
UP (constr.)	0.68	0.68	0.67±0.02	0.99	0.99	0.98±0.01	0.46	0.47	0.45±0.08	0.89	0.97	0.84±0.09	0.00	0.00	0.00±0.00	0.68	0.69	0.66±0.02
UP (fixed α)	0.64	0.65	0.64±0.04	0.99	0.99	0.99±0.00	0.49	0.49	0.49±0.04	0.70	0.95	0.77±0.11	0.00	0.00	0.00±0.00	0.64	0.69	0.65±0.02
EP	0.68	1.00	0.71±0.25	1.00	0.97	0.74±0.17	0.55	0.85	0.47±0.28	0.92	0.97	0.77±0.21	0.00	0.00	0.00±0.00	0.70	0.80	0.63±0.09
EP (constr.)	0.67	0.79	0.68±0.07	0.99	1.00	0.99±0.01	0.47	0.85	0.53±0.26	0.77	0.92	0.84±0.09	0.00	0.00	0.00±0.00	0.66	0.76	0.68±0.05
EP (fixed α)	0.64	0.65	0.65±0.03	0.99	0.99	0.99±0.01	0.00	0.00	0.00±0.00	0.92	0.92	0.79±0.08	0.00	0.00	0.00±0.00	0.60	0.60	0.58±0.02
EP2	0.68	0.65	0.63±0.07	0.98	1.00	0.96±0.08	0.50	0.51	0.37±0.20	0.74	0.77	0.85±0.12	0.00	0.84	0.01±0.09	0.66	0.81	0.64±0.04
EP2 (constr.)	0.64	0.64	0.65±0.02	0.99	0.99	0.99±0.00	0.49	0.52	0.51±0.05	0.83	0.90	0.82±0.03	0.00	0.00	0.00±0.00	0.67	0.68	0.67±0.01
EP2 (fixed α)	0.64	0.65	0.65±0.03	0.99	0.99	0.99±0.01	0.49	0.49	0.48±0.04	0.83	0.95	0.87±0.06	0.00	0.00	0.00±0.00	0.67	0.69	0.67±0.02
HUP	0.68	0.69	0.67±0.03	0.99	0.99	0.99±0.01	0.51	0.51	0.51±0.02	0.92	0.92	0.92±0.04	0.00	0.00	0.00±0.00	0.69	0.69	0.69±0.02
HUP (constr.)	0.64	0.64	0.64±0.03	0.99	0.99	0.99±0.00	0.48	0.49	0.48±0.04	0.68	0.77	0.70±0.05	0.00	0.72	0.04±0.17	0.64	0.78	0.65±0.04
K	0.56	0.79	0.56±0.04	0.43	0.58	0.44±0.06	0.41	0.57	0.42±0.05	0.48	0.69	0.55±0.11	0.67	0.00	0.70±0.06	0.03	0.55	0.05±0.05
Prop.	0.67	0.68	0.67±0.03	0.73	0.99	0.87±0.13	0.50	0.50	0.50±0.03	0.92	0.95	0.92±0.04	0.62	0.71	0.65±0.06	0.10	0.71	0.13±0.06
Oh	0.43	0.64	0.66±0.16	0.31	0.96	0.63±0.17	0.31	0.56	0.40±0.07	0.62	0.77	0.75±0.14	0.71	0.00	0.55±0.22	0.14	0.67	0.24±0.09

Table D.14: Target scores for the results from the MCMC sampler on the five target data set. The individual moves were allowed to propose remaining in the current state.

Prior	LL			Score			K			α				λ		
	MAP	Max Score	Truth	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Truth	Avg.	MAP	Max Score	Avg.
DP	-15319.1	-15329.5	-15340.8	0.67	0.68	0.67±0.00	4.0	4.0	4.0±0.00	0.578	3.13	0.634	0.906±0.40	14.3	14.8	14.6±0.38
DP (constr.)	-15337.1	-15345.6	-15344.4	0.35	0.35	0.35±0.00	13.0	13.0	13.0±0.00	2.34	1.82	0.122	2.54±0.75	13.4	13.9	13.6±0.40
DP (fixed α)	-15380.0	-15386.6	-15389.2	0.57	0.59	0.57±0.00	3.0	3.0	3.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.8	14.5	14.8±0.51
UP	-15261.0	-15266.7	-15303.3	0.81	0.83	0.81±0.00	5.0	5.0	5.0±0.00	0.0547	0.194	0.0733	0.095±0.04	13.9	14.5	14.9±0.70
UP (constr.)	-15435.0	-15442.8	-15307.2	0.44	0.44	0.44±0.00	2.0	2.0	2.0±0.00	0.00724	0.000259	0.0118	0.00692±0.00	15.3	15.3	15.4±0.36
UP (fixed α)	-15338.3	-15351.2	-15344.1	0.48	0.48	0.48±0.00	9.0	9.0	9.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.5	13.0	14.1±0.54
EP	-15426.3	-15431.3	-15388.8	0.57	0.57	0.57±0.00	3.0	3.0	3.0±0.00	0.00314	0.0118	0.014	0.00218±0.00	14.7	14.5	14.8±0.60
EP (constr.)	-15443.0	-15443.9	-15389.1	0.58	0.58	0.58±0.00	3.0	3.0	3.0±0.00	0.00217	0.00396	0.0151	0.00346±0.00	14.9	15.3	14.8±0.49
EP (fixed α)	-15431.9	-15432.8	-15419.7	0.55	0.55	0.51±0.01	8.0	8.0	8.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	13.9	14.6	14.7±0.34
EP2	-15357.5	-15369.0	-15346.5	0.57	0.57	0.57±0.00	3.0	3.0	3.0±0.00	0.115	1.43	1.9	0.0621±0.06	14.7	14.5	14.6±0.15
EP2 (constr.)	-15327.7	-15338.1	-15340.5	0.51	0.51	0.51±0.01	9.0	9.0	9.0±0.00	0.41	0.187	0.119	0.978±1.61	13.8	14.2	14.0±0.27
EP2 (fixed α)	-15368.1	-15374.5	-15382.1	0.57	0.57	0.57±0.00	3.0	3.0	3.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.7	15.6	15.0±0.54
HUP	-15305.4	-15307.2	-15337.1	0.81	0.82	0.81±0.00	5.0	5.0	5.0±0.00	N/A	N/A	N/A	N/A	14.2	14.0	14.2±0.09
HUP (constr.)	-15401.1	-15407.9	-15357.9	0.43	0.44	0.43±0.00	10.0	10.0	10.0±0.00	N/A	N/A	N/A	N/A	13.9	13.0	13.9±0.48
K	-15236.4	-15237.2	-15157.1	0.57	0.57	0.57±0.00	7.0	7.0	7.0±0.00	N/A	N/A	N/A	N/A	13.5	12.9	13.6±0.46
Prop.	-15359.2	-15366.1	-15388.7	0.62	0.63	0.61±0.00	7.0	7.0	7.0±0.00	N/A	N/A	N/A	N/A	14.3	14.0	14.2±0.17
Oh	-15377.8	-15378.0	-15418.1	0.74	0.74	0.74±0.00	5.0	5.0	5.0±0.00	N/A	N/A	N/A	N/A	14.1	13.9	14.1±0.49

Table D.15: Summary of results from the PMC sampler on the five target data set.

Prior	Target Score																	
	τ_1			τ_2			τ_3			τ_4			τ_5			Combined		
	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.
DP	0.64	0.64	0.64±0.00	0.97	0.99	0.97±0.00	0.50	0.53	0.50±0.01	0.87	0.87	0.87±0.00	0.00	0.00	0.00±0.00	0.67	0.68	0.67±0.00
DP (constr.)	0.60	0.60	0.60±0.00	0.98	0.98	0.98±0.00	0.60	0.60	0.60±0.00	0.92	0.92	0.92±0.00	0.67	0.67	0.67±0.00	0.35	0.35	0.35±0.00
DP (fixed α)	0.64	0.64	0.64±0.00	0.97	0.97	0.96±0.00	0.00	0.00	0.00±0.00	0.79	0.87	0.79±0.02	0.00	0.00	0.00±0.00	0.57	0.59	0.57±0.00
UP	0.64	0.64	0.64±0.00	0.97	0.99	0.97±0.00	0.51	0.54	0.51±0.01	0.92	0.92	0.90±0.03	0.75	0.80	0.77±0.02	0.81	0.83	0.81±0.00
UP (constr.)	0.64	0.65	0.64±0.00	0.99	0.99	0.99±0.00	0.00	0.00	0.00±0.00	0.00	0.00	0.00±0.00	0.00	0.00	0.00±0.00	0.44	0.44	0.44±0.00
UP (fixed α)	0.43	0.43	0.43±0.00	0.98	0.98	0.98±0.00	0.50	0.50	0.50±0.00	0.95	0.95	0.95±0.00	0.80	0.80	0.80±0.00	0.48	0.48	0.48±0.00
EP	0.65	0.65	0.65±0.00	0.96	0.96	0.96±0.00	0.00	0.00	0.00±0.00	0.77	0.77	0.77±0.00	0.00	0.00	0.00±0.00	0.57	0.57	0.57±0.00
EP (constr.)	0.64	0.64	0.64±0.00	0.98	0.98	0.98±0.00	0.00	0.00	0.00±0.00	0.80	0.80	0.80±0.00	0.00	0.00	0.00±0.00	0.58	0.58	0.58±0.00
EP (fixed α)	0.89	0.89	0.66±0.05	0.68	0.68	0.68±0.00	0.59	0.59	0.47±0.03	0.81	0.81	0.81±0.00	0.78	0.78	0.78±0.00	0.55	0.55	0.51±0.01
EP2	0.64	0.64	0.64±0.00	0.99	0.99	0.99±0.00	0.00	0.00	0.00±0.00	0.72	0.77	0.75±0.00	0.00	0.00	0.00±0.00	0.57	0.57	0.57±0.00
EP2 (constr.)	0.88	0.88	0.88±0.00	0.98	0.98	0.95±0.08	0.60	0.60	0.60±0.00	0.86	0.86	0.86±0.00	0.80	0.80	0.80±0.00	0.51	0.51	0.51±0.01
EP2 (fixed α)	0.64	0.64	0.64±0.00	0.99	1.00	0.99±0.01	0.00	0.00	0.00±0.00	0.72	0.72	0.72±0.00	0.00	0.00	0.00±0.00	0.57	0.57	0.57±0.00
HUP	0.64	0.64	0.64±0.00	0.97	0.97	0.97±0.00	0.51	0.48	0.51±0.00	0.89	0.89	0.89±0.00	0.78	0.90	0.78±0.02	0.81	0.82	0.81±0.00
HUP (constr.)	0.64	0.64	0.64±0.00	0.96	0.96	0.96±0.00	0.48	0.48	0.48±0.00	0.80	0.94	0.80±0.01	0.75	0.75	0.75±0.00	0.43	0.44	0.43±0.00
K	0.64	0.64	0.64±0.00	0.96	0.96	0.96±0.00	0.44	0.44	0.44±0.00	0.75	0.75	0.75±0.00	0.69	0.69	0.69±0.00	0.57	0.57	0.57±0.00
Prop.	0.68	0.68	0.68±0.00	0.99	0.99	0.99±0.00	0.48	0.50	0.47±0.01	0.95	0.95	0.90±0.01	0.84	0.84	0.84±0.00	0.62	0.63	0.61±0.00
Oh	0.64	0.64	0.64±0.00	0.96	0.96	0.96±0.00	0.49	0.49	0.49±0.00	0.75	0.75	0.75±0.00	0.54	0.54	0.54±0.00	0.74	0.74	0.74±0.00

Table D.16: Target scores for the results from the PMC sampler on the five target data set.

Prior	LL			Score			K			α				λ		
	MAP	Max Score	Truth	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Truth	Avg.	MAP	Max Score	Avg.
DP	-15305.7	-15305.7	-15340.8	0.68	0.68	0.68±0.00	6.0	6.0	6.0±0.00	3.63	3.63	0.634	3.63±0.00	14.2	14.2	14.2±0.00
DP (constr.)	-15397.1	-15397.6	-15344.4	0.68	0.68	0.66±0.03	6.0	6.0	6.2±0.43	1.03	1.06	0.122	1.33±0.55	14.3	13.9	14.3±0.49
DP (fixed α)	-15416.1	-15416.5	-15389.2	0.60	0.60	0.60±0.00	3.0	3.0	3.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.9	14.5	14.9±0.48
UP	-15388.4	-15389.1	-15303.3	0.67	0.67	0.67±0.00	4.0	4.0	4.0±0.00	0.052	0.0245	0.0733	0.0712±0.03	14.5	14.3	14.5±0.47
UP (constr.)	-15387.8	-15390.4	-15307.2	0.77	0.77	0.77±0.00	5.0	5.0	5.0±0.00	0.0718	0.142	0.0118	0.0915±0.04	14.3	13.5	14.3±0.49
UP (fixed α)	-15350.4	-15353.2	-15344.1	0.57	0.58	0.57±0.00	3.0	3.0	3.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	15.3	14.8	15.3±0.05
EP	-15398.9	-15401.5	-15388.8	0.65	0.65	0.65±0.00	4.0	4.0	4.0±0.00	0.00317	0.00838	0.014	0.00321±0.00	14.4	13.7	14.5±0.37
EP (constr.)	-15379.1	-15384.4	-15389.1	0.58	0.58	0.58±0.00	3.0	3.0	3.0±0.00	0.00225	0.00419	0.0151	0.00331±0.00	14.9	15.3	14.8±0.41
EP (fixed α)	-15443.3	-15443.7	-15419.7	0.59	0.59	0.59±0.00	3.0	3.0	3.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.9	14.5	14.9±0.49
EP2	-15428.9	-15430.7	-15346.5	0.57	0.57	0.57±0.00	3.0	3.0	3.0±0.00	0.103	0.233	1.9	0.106±0.07	14.8	14.1	15.1±0.46
EP2 (constr.)	-15295.8	-15304.4	-15340.5	0.65	0.65	0.65±0.00	4.0	4.0	4.0±0.00	0.155	0.247	0.119	0.18±0.04	14.1	14.2	14.4±0.61
EP2 (fixed α)	-15328.1	-15333.0	-15382.1	0.67	0.67	0.66±0.01	4.0	4.0	4.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.7	14.4	14.4±0.20
HUP	-15357.2	-15360.5	-15337.1	0.59	0.60	0.59±0.00	3.0	3.0	3.0±0.00	N/A	N/A	N/A	N/A	14.5	15.6	14.5±0.15
HUP (constr.)	-15408.9	-15409.6	-15357.9	0.65	0.79	0.75±0.06	4.0	5.0	4.7±0.49	N/A	N/A	N/A	N/A	14.4	14.2	14.2±0.48
K	-15169.0	-15170.2	-15157.1	0.74	0.74	0.74±0.00	5.0	5.0	5.0±0.00	N/A	N/A	N/A	N/A	14.1	13.3	14.1±0.48
Prop.	-15378.1	-15380.4	-15388.7	0.39	0.39	0.39±0.00	11.0	11.0	11.0±0.00	N/A	N/A	N/A	N/A	13.9	13.9	13.7±0.52
Oh	-15294.6	-15296.1	-15418.1	0.77	0.78	0.77±0.00	5.0	5.0	5.0±0.00	N/A	N/A	N/A	N/A	14.0	13.8	13.8±0.56

Table D.17: Summary of results from the PMC sampler on the five target data set. The individual moves were allowed to propose remaining in the current state.

Prior	Target Score																	
	τ_1			τ_2			τ_3			τ_4			τ_5			Combined		
	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.
DP	0.68	0.68	0.68±0.00	0.99	0.99	0.99±0.00	0.49	0.49	0.49±0.00	0.92	0.92	0.92±0.00	0.67	0.67	0.67±0.00	0.68	0.68	0.68±0.00
DP (constr.)	0.60	0.60	0.60±0.00	0.98	0.98	0.98±0.00	0.52	0.52	0.52±0.00	0.92	0.92	0.92±0.00	0.67	0.67	0.68±0.02	0.68	0.68	0.66±0.03
DP (fixed α)	0.64	0.64	0.64±0.00	0.98	0.98	0.98±0.00	0.00	0.00	0.00±0.00	0.92	0.92	0.92±0.00	0.00	0.00	0.00±0.00	0.60	0.60	0.60±0.00
UP	0.65	0.65	0.65±0.00	0.98	0.98	0.98±0.00	0.45	0.45	0.45±0.00	0.92	0.92	0.92±0.00	0.00	0.00	0.00±0.00	0.67	0.67	0.67±0.00
UP (constr.)	0.65	0.65	0.65±0.00	0.98	0.98	0.98±0.00	0.45	0.45	0.45±0.00	0.92	0.92	0.92±0.00	0.60	0.60	0.60±0.00	0.77	0.77	0.77±0.00
UP (fixed α)	0.64	0.64	0.64±0.00	0.99	0.99	0.99±0.00	0.00	0.00	0.00±0.00	0.74	0.80	0.74±0.01	0.00	0.00	0.00±0.00	0.57	0.58	0.57±0.00
EP	0.65	0.65	0.65±0.00	0.96	0.96	0.96±0.00	0.46	0.46	0.46±0.00	0.77	0.77	0.77±0.00	0.00	0.00	0.00±0.00	0.65	0.65	0.65±0.00
EP (constr.)	0.68	0.68	0.68±0.00	0.99	1.00	0.99±0.00	0.00	0.00	0.00±0.00	0.76	0.76	0.75±0.01	0.00	0.00	0.00±0.00	0.58	0.58	0.58±0.00
EP (fixed α)	0.64	0.64	0.64±0.00	0.97	0.97	0.97±0.00	0.00	0.00	0.00±0.00	0.92	0.92	0.92±0.00	0.00	0.00	0.00±0.00	0.59	0.59	0.59±0.00
EP2	0.64	0.64	0.64±0.00	0.98	0.98	0.98±0.00	0.00	0.00	0.00±0.00	0.75	0.75	0.75±0.00	0.00	0.00	0.00±0.00	0.57	0.57	0.57±0.00
EP2 (constr.)	0.64	0.65	0.65±0.01	0.99	1.00	0.99±0.00	0.48	0.48	0.48±0.00	0.72	0.72	0.72±0.00	0.00	0.00	0.00±0.00	0.65	0.65	0.65±0.00
EP2 (fixed α)	0.64	0.64	0.64±0.00	0.99	1.00	0.99±0.00	0.49	0.49	0.49±0.00	0.83	0.83	0.76±0.04	0.00	0.00	0.00±0.00	0.67	0.67	0.66±0.01
HUP	0.64	0.64	0.64±0.00	0.99	0.99	0.99±0.00	0.00	0.00	0.00±0.00	0.87	0.92	0.87±0.01	0.00	0.00	0.00±0.00	0.59	0.60	0.59±0.00
HUP (constr.)	0.64	0.64	0.64±0.00	0.98	0.98	0.98±0.00	0.50	0.50	0.50±0.00	0.75	0.75	0.75±0.00	0.00	0.78	0.56±0.35	0.65	0.79	0.75±0.06
K	0.64	0.64	0.64±0.00	0.97	0.97	0.97±0.00	0.48	0.48	0.48±0.00	0.73	0.73	0.73±0.00	0.56	0.56	0.56±0.00	0.74	0.74	0.74±0.00
Prop.	0.56	0.56	0.56±0.00	0.99	0.99	0.99±0.00	0.51	0.51	0.51±0.00	0.95	0.95	0.95±0.00	0.62	0.62	0.63±0.00	0.39	0.39	0.39±0.00
Oh	0.64	0.64	0.66±0.01	0.99	0.99	0.99±0.00	0.49	0.49	0.48±0.00	0.72	0.74	0.73±0.01	0.72	0.72	0.72±0.00	0.77	0.78	0.77±0.00

Table D.18: Target scores for the results from the PMC sampler on the five target data set. The individual moves were allowed to propose remaining in the current state.

Prior	LL			Score			K			α				λ		
	MAP	Max Score	Truth	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Truth	Avg.	MAP	Max Score	Avg.
DP	-15369.4	-15369.4	-15340.8	0.67	0.67	0.67±0.00	4.0	4.0	4.0±0.00	3.36	3.36	0.634	3.36±0.00	13.8	13.8	13.8±0.00
DP (constr.)	-15365.8	-15365.8	-15344.4	0.58	0.58	0.58±0.00	3.0	3.0	3.0±0.00	0.0345	0.0345	0.122	0.0345±0.00	15.5	15.5	15.5±0.00
DP (fixed α)	-15405.6	-15405.6	-15389.2	0.60	0.60	0.60±0.00	3.0	3.0	3.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	15.2	15.2	15.2±0.25
UP	-15333.8	-15333.8	-15303.3	0.65	0.65	0.65±0.00	4.0	4.0	4.0±0.00	0.133	0.133	0.0733	0.133±0.00	13.7	13.7	13.7±0.00
UP (constr.)	-15349.0	-15349.0	-15307.2	0.66	0.66	0.66±0.00	4.0	4.0	4.0±0.00	0.00928	0.00928	0.0118	0.00931±0.00	15.1	15.1	15.1±0.01
UP (fixed α)	-15345.6	-15345.6	-15344.1	0.66	0.66	0.66±0.00	4.0	4.0	4.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	13.7	13.7	13.7±0.00
EP	-15449.2	-15449.2	-15388.8	0.60	0.60	0.57±0.01	3.0	3.0	3.0±0.00	0.0285	0.0285	0.014	0.00324±0.01	14.6	14.6	15.8±0.39
EP (constr.)	-15431.4	-15431.4	-15389.1	0.58	0.58	0.58±0.00	3.0	3.0	3.0±0.00	0.0104	0.0104	0.0151	0.0104±0.00	14.8	14.8	14.8±0.00
EP (fixed α)	-15444.8	-15444.8	-15419.7	0.59	0.59	0.59±0.00	3.0	3.0	3.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	14.9	14.9	14.9±0.00
EP2	-15365.5	-15365.5	-15346.5	0.65	0.65	0.65±0.00	4.0	4.0	4.0±0.00	1.32	1.32	1.9	1.32±0.00	15.0	15.0	15.0±0.00
EP2 (constr.)	-15386.2	-15390.9	-15340.5	0.55	0.57	0.56±0.00	3.0	3.0	3.0±0.00	0.11	0.163	0.119	0.112±0.01	14.2	15.1	14.2±0.19
EP2 (fixed α)	-15407.2	-15407.2	-15382.1	0.56	0.56	0.56±0.00	3.0	3.0	3.0±0.00	1e-06	1e-06	1e-06	1e-06±0.00	15.4	15.4	15.4±0.00
HUP	-15371.9	-15371.9	-15337.1	0.68	0.68	0.68±0.00	4.0	4.0	4.0±0.00	N/A	N/A	N/A	N/A	14.3	14.3	14.3±0.00
HUP (constr.)	-15401.7	-15401.7	-15357.9	0.57	0.57	0.57±0.00	3.0	3.0	3.0±0.00	N/A	N/A	N/A	N/A	14.6	14.6	14.6±0.00
K	-15232.2	-15232.2	-15157.1	0.63	0.63	0.63±0.00	6.0	6.0	6.0±0.00	N/A	N/A	N/A	N/A	13.9	13.9	13.9±0.00
Prop.	-15416.2	-15416.2	-15388.7	0.70	0.70	0.70±0.00	4.0	4.0	4.0±0.00	N/A	N/A	N/A	N/A	15.2	15.2	15.2±0.21
Oh	-15410.2	-15413.7	-15418.1	0.56	0.57	0.56±0.00	3.0	3.0	3.0±0.00	N/A	N/A	N/A	N/A	14.8	14.4	14.8±0.02

Table D.19: Summary of results from the PMC sampler, using the k-add mixture proposal, on the five target data set.

Prior	Target Score																	
	τ_1			τ_2			τ_3			τ_4			τ_5			Combined		
	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.	MAP	Max Score	Avg.
DP	0.64	0.64	0.64±0.00	0.98	0.98	0.98±0.00	0.46	0.46	0.46±0.00	0.92	0.92	0.92±0.00	0.00	0.00	0.00±0.00	0.67	0.67	0.67±0.00
DP (constr.)	0.64	0.64	0.64±0.00	0.99	0.99	0.99±0.00	0.00	0.00	0.00±0.00	0.83	0.83	0.83±0.00	0.00	0.00	0.00±0.00	0.58	0.58	0.58±0.00
DP (fixed α)	0.65	0.65	0.65±0.00	0.99	0.99	0.99±0.01	0.00	0.00	0.00±0.00	0.95	0.95	0.95±0.00	0.00	0.00	0.00±0.00	0.60	0.60	0.60±0.00
UP	0.53	0.53	0.53±0.00	0.97	0.97	0.97±0.00	0.48	0.48	0.48±0.00	0.88	0.88	0.88±0.00	0.00	0.00	0.00±0.00	0.65	0.65	0.65±0.00
UP (constr.)	0.64	0.64	0.64±0.00	0.98	0.98	0.98±0.00	0.51	0.51	0.51±0.00	0.81	0.81	0.81±0.00	0.00	0.00	0.00±0.00	0.66	0.66	0.66±0.00
UP (fixed α)	0.64	0.64	0.64±0.00	0.96	0.96	0.96±0.00	0.48	0.48	0.48±0.00	0.82	0.82	0.82±0.00	0.00	0.00	0.00±0.00	0.66	0.66	0.66±0.00
EP	0.63	0.63	0.65±0.02	0.98	0.98	0.96±0.01	0.00	0.00	0.00±0.00	0.95	0.95	0.78±0.05	0.00	0.00	0.00±0.00	0.60	0.60	0.57±0.01
EP (constr.)	0.64	0.64	0.64±0.00	0.99	0.99	0.99±0.00	0.00	0.00	0.00±0.00	0.78	0.78	0.78±0.00	0.00	0.00	0.00±0.00	0.58	0.58	0.58±0.00
EP (fixed α)	0.64	0.64	0.64±0.00	0.99	0.99	0.99±0.00	0.00	0.00	0.00±0.00	0.89	0.89	0.89±0.00	0.00	0.00	0.00±0.00	0.59	0.59	0.59±0.00
EP2	0.65	0.65	0.65±0.00	0.97	0.97	0.97±0.00	0.48	0.48	0.48±0.00	0.79	0.79	0.79±0.00	0.00	0.00	0.00±0.00	0.65	0.65	0.65±0.00
EP2 (constr.)	0.64	0.65	0.64±0.00	0.96	1.00	0.96±0.01	0.00	0.00	0.00±0.00	0.70	0.72	0.70±0.01	0.00	0.00	0.00±0.00	0.55	0.57	0.56±0.00
EP2 (fixed α)	0.65	0.65	0.65±0.00	0.97	0.97	0.97±0.00	0.00	0.00	0.00±0.00	0.70	0.70	0.70±0.00	0.00	0.00	0.00±0.00	0.56	0.56	0.56±0.00
HUP	0.64	0.64	0.64±0.00	0.96	0.96	0.96±0.00	0.49	0.49	0.49±0.00	0.92	0.92	0.92±0.00	0.00	0.00	0.00±0.00	0.68	0.68	0.68±0.00
HUP (constr.)	0.64	0.64	0.64±0.00	0.99	0.99	0.99±0.00	0.00	0.00	0.00±0.00	0.77	0.77	0.77±0.00	0.00	0.00	0.00±0.00	0.57	0.57	0.57±0.00
K	0.64	0.64	0.64±0.00	0.98	0.98	0.98±0.00	0.48	0.48	0.48±0.00	0.73	0.73	0.73±0.00	0.52	0.52	0.52±0.00	0.63	0.63	0.63±0.00
Prop.	0.67	0.67	0.67±0.00	1.00	1.00	1.00±0.00	0.50	0.50	0.50±0.00	0.97	0.97	0.97±0.01	0.00	0.00	0.00±0.00	0.70	0.70	0.70±0.00
Oh	0.64	0.64	0.64±0.00	0.98	0.96	0.98±0.00	0.00	0.00	0.00±0.00	0.72	0.77	0.72±0.00	0.00	0.00	0.00±0.00	0.56	0.57	0.56±0.00

Table D.20: Target scores for the results from the PMC sampler, using the k-add mixture proposal, on the five target data set.

Prior	LL		K		α		λ	
	MAP	Avg.	MAP	Avg.	MAP	Avg.	MAP	Avg.
DP	-4607.7	-4654.2	50.0	44.8±7.69	15.4	17.3±11.99	0.6	0.8±0.27
DP (constr.)	-4605.7	-4653.9	47.0	46.4±9.80	19.1	15.2±10.99	0.9	0.8±0.33
DP (fixed α)	-4802.5	-4810.0	5.0	5.8±0.74	1e-06	1e-06±0.00	1.9	1.9±0.14
UP	-4603.7	-4656.5	60.0	49.7±14.71	5.53	7.2±4.96	0.6	0.8±0.34
UP (constr.)	-4597.7	-4664.0	56.0	49.4±10.76	3.83	5.45±3.70	0.5	0.7±0.27
UP (fixed α)	-4743.0	-4763.8	7.0	9.2±1.26	1e-06	1e-06±0.00	1.7	1.6±0.15
EP	-4596.3	-4671.0	63.0	58.8±12.12	3.56	2.9±2.10	0.7	0.7±0.29
EP (constr.)	-4576.1	-4668.4	66.0	58.6±12.24	3.2	2.12±1.55	0.5	0.7±0.30
EP (fixed α)	-4813.7	-4824.4	7.0	7.0±0.68	1e-06	1e-06±0.00	1.6	1.6±0.14
EP2	-4612.9	-4671.5	52.0	49.2±10.43	255	208±53.16	0.7	0.8±0.23
EP2 (constr.)	-4641.4	-4697.1	56.0	48.2±11.47	254	182±63.77	0.7	0.8±0.28
EP2 (fixed α)	-4788.7	-4798.0	7.0	6.8±0.70	1e-06	1e-06±0.00	1.6	1.7±0.14
HUP	-4492.9	-4543.5	56.0	49.3±8.29	N/A	N/A	0.5	0.7±0.28
HUP (constr.)	-4716.3	-4752.6	41.0	33.7±6.99	N/A	N/A	0.9	1.0±0.18
K	-3873.4	-3966.9	54.0	51.9±6.95	N/A	N/A	0.4	0.5±0.18
Prop.	-4856.2	-4875.3	19.0	23.6±3.69	N/A	N/A	1.3	1.2±0.23
Oh	-4947.8	-4957.2	6.0	6.3±1.83	N/A	N/A	2.4	2.4±0.14

Table D.21: Summary of results from the MCMC sampler on the field test data set.

Prior	LL		K		α		λ	
	MAP	Avg.	MAP	Avg.	MAP	Avg.	MAP	Avg.
DP	-4767.8	-4797.5	3.0	2.7±0.77	0.413	1.74±2.39	2.2	2.2±0.15
DP (constr.)	-4790.6	-4795.5	2.0	2.0±0.15	0.0432	0.0271±0.15	2.3	2.3±0.14
DP (fixed α)	-4834.4	-4835.5	1.0	1.0±0.00	1e-06	1e-06±0.00	2.4	2.4±0.14
UP	-4747.4	-4796.2	4.0	3.2±0.58	0.0659	0.409±0.66	1.9	2.1±0.15
UP (constr.)	-4741.7	-4764.7	3.0	2.6±0.50	0.0199	0.015±0.08	2.1	2.2±0.16
UP (fixed α)	-4787.0	-4791.1	2.0	2.0±0.10	1e-06	1e-06±0.00	2.2	2.3±0.14
EP	-4834.4	-4848.6	1.0	1.0±0.00	1.78e-05	0.0101±0.04	2.4	2.4±0.14
EP (constr.)	-4834.4	-4848.6	1.0	1.0±0.00	1.78e-05	0.0101±0.04	2.4	2.4±0.14
EP (fixed α)	-4834.4	-4835.5	1.0	1.0±0.00	1e-06	1e-06±0.00	2.4	2.4±0.14
EP2	-4837.8	-4842.9	1.0	1.0±0.00	0.000884	0.0694±0.07	2.4	2.4±0.14
EP2 (constr.)	-4837.8	-4842.9	1.0	1.0±0.00	0.000884	0.0694±0.07	2.4	2.4±0.14
EP2 (fixed α)	-4834.4	-4835.5	1.0	1.0±0.00	1e-06	1e-06±0.00	2.4	2.4±0.14
HUP	-4787.5	-4792.3	2.0	2.0±0.15	N/A	N/A	2.2	2.3±0.14
HUP (constr.)	-4704.5	-4726.4	31.0	24.1±5.38	N/A	N/A	1.0	1.0±0.16
K	-3768.9	-3870.9	52.0	48.7±7.65	N/A	N/A	0.4	0.6±0.20
Prop.	-4844.1	-4869.3	21.0	24.6±3.56	N/A	N/A	1.4	1.3±0.20
Oh	-4920.6	-4930.8	6.0	6.9±1.39	N/A	N/A	2.3	2.4±0.14

Table D.22: Summary of results from the MCMC sampler on the field test data set. The individual moves were allowed to propose remaining in the current state.

Prior	LL		K		α		λ	
	MAP	Avg.	MAP	Avg.	MAP	Avg.	MAP	Avg.
DP	-4711.6	-4713.5	18.0	18.0±0.00	4.9	4.57±1.11	1.5	1.4±0.11
DP (constr.)	-4714.9	-4717.5	18.0	18.0±0.00	3.94	5.04±1.08	1.4	1.4±0.12
DP (fixed α)	-4797.1	-4798.5	2.0	2.0±0.00	1e-06	1e-06±0.00	2.2	2.1±0.10
UP	-4654.3	-4657.1	8.0	8.0±0.00	0.198	0.195±0.11	1.7	1.6±0.09
UP (constr.)	-4671.9	-4672.0	6.0	6.0±0.00	0.138	0.137±0.01	1.7	1.7±0.02
UP (fixed α)	-4731.9	-4731.9	4.0	4.0±0.00	1e-06	1e-06±0.00	1.9	1.9±0.00
EP	-4764.2	-4765.8	9.0	9.0±0.00	0.0385	0.0374±0.01	1.6	1.7±0.06
EP (constr.)	-4797.5	-4799.8	3.0	3.0±0.00	0.00259	0.00497±0.00	2.1	2.1±0.13
EP (fixed α)	-4814.4	-4816.4	8.0	8.0±0.00	1e-06	1e-06±0.00	1.7	1.6±0.07
EP2	-4768.5	-4768.7	7.0	7.0±0.00	0.642	0.684±0.14	1.6	1.6±0.04
EP2 (constr.)	-4707.2	-4707.6	8.0	8.0±0.00	1.97	1.89±0.21	1.6	1.6±0.02
EP2 (fixed α)	-4755.0	-4758.7	6.0	6.0±0.00	1e-06	1e-06±0.00	1.7	1.6±0.08
HUP	-4745.0	-4745.6	8.0	8.0±0.00	N/A	N/A	1.8	1.8±0.13
HUP (constr.)	-4750.4	-4752.3	9.0	9.0±0.00	N/A	N/A	1.6	1.5±0.06
K	-4164.2	-4164.7	19.0	19.0±0.00	N/A	N/A	0.7	0.8±0.08
Prop.	-4891.7	-4894.9	9.0	9.0±0.00	N/A	N/A	1.8	1.9±0.10
Oh	-4926.9	-4930.1	4.0	4.9±0.76	N/A	N/A	2.4	2.4±0.15

Table D.23: Summary of results from the PMC sampler on the field test data set.

Prior	LL		K		α		λ	
	MAP	Avg.	MAP	Avg.	MAP	Avg.	MAP	Avg.
DP	-4714.0	-4717.8	11.0	11.0±0.00	3.53	3.08±1.17	1.8	1.8±0.07
DP (constr.)	-4715.3	-4719.0	10.0	10.0±0.00	1.9	2.23±0.77	1.8	1.7±0.14
DP (fixed α)	-4798.2	-4801.5	2.0	2.0±0.00	1e-06	1e-06±0.00	2.3	2.3±0.11
UP	-4674.9	-4681.4	6.0	6.0±0.00	0.103	0.131±0.03	1.8	1.6±0.06
UP (constr.)	-4674.3	-4676.2	6.0	6.0±0.00	0.0901	0.113±0.05	1.7	1.7±0.11
UP (fixed α)	-4757.8	-4770.2	3.0	3.0±0.00	1e-06	1e-06±0.00	2.1	2.1±0.05
EP	-4822.3	-4822.3	3.0	3.0±0.00	0.00455	0.00455±0.00	2.2	2.2±0.00
EP (constr.)	-4802.0	-4813.0	3.0	3.0±0.00	0.00506	0.00273±0.00	2.0	2.2±0.02
EP (fixed α)	-4834.4	-4835.1	1.0	1.0±0.00	1e-06	1e-06±0.00	2.4	2.4±0.14
EP2	-4732.3	-4733.8	5.0	5.0±0.00	1.02	1.05±0.19	1.9	1.7±0.09
EP2 (constr.)	-4710.4	-4717.6	8.0	8.0±0.00	0.764	0.924±0.57	1.5	1.6±0.08
EP2 (fixed α)	-4755.2	-4756.0	6.0	6.0±0.00	1e-06	1e-06±0.00	1.8	1.8±0.08
HUP	-4703.5	-4703.5	8.0	8.0±0.00	N/A	N/A	1.8	1.8±0.00
HUP (constr.)	-4741.4	-4743.8	8.0	8.0±0.02	N/A	N/A	1.6	1.6±0.13
K	-4051.7	-4052.3	25.0	25.0±0.00	N/A	N/A	0.7	0.6±0.05
Prop.	-4885.9	-4887.5	8.0	8.0±0.00	N/A	N/A	2.1	2.1±0.13
Oh	-4926.8	-4929.2	4.0	4.4±0.55	N/A	N/A	2.4	2.4±0.14

Table D.24: Summary of results from the PMC sampler on the field test data set. The individual moves were allowed to propose remaining in the current state.

Prior	LL		K		α		λ	
	MAP	Avg.	MAP	Avg.	MAP	Avg.	MAP	Avg.
DP	-4796.8	-4806.4	2.0	3.0±0.00	0.393	3±0.01	1.9	2.1±0.00
DP (constr.)	-4794.8	-4795.3	2.0	2.0±0.00	0.0255	0.0254±0.00	2.0	2.0±0.08
DP (fixed α)	-4834.4	-4835.2	1.0	1.0±0.00	1e-06	1e-06±0.00	2.4	2.4±0.15
UP	-4769.6	-4772.0	3.0	3.2±0.39	0.202	0.103±0.18	2.0	2.1±0.03
UP (constr.)	-4761.8	-4761.8	3.0	3.0±0.00	0.0353	0.0354±0.00	1.9	1.9±0.01
UP (fixed α)	-4785.9	-4786.5	2.0	2.0±0.00	1e-06	1e-06±0.00	2.3	2.3±0.05
EP	-4834.6	-4836.3	1.0	1.0±0.00	3.75e-05	0.000594±0.00	2.5	2.4±0.15
EP (constr.)	-4834.7	-4836.7	1.0	1.0±0.00	0.000157	0.000523±0.00	2.4	2.5±0.21
EP (fixed α)	-4834.4	-4835.1	1.0	1.0±0.00	1e-06	1e-06±0.00	2.4	2.4±0.14
EP2	-4796.3	-4796.3	3.0	3.0±0.00	1.81	1.81±0.00	2.1	2.1±0.00
EP2 (constr.)	-4837.0	-4839.8	1.0	1.0±0.00	0.00151	0.0378±0.03	2.4	2.4±0.15
EP2 (fixed α)	-4834.4	-4835.2	1.0	1.0±0.00	1e-06	1e-06±0.00	2.4	2.4±0.14
HUP	-4788.4	-4793.1	2.0	2.0±0.00	N/A	N/A	2.1	2.2±0.11
HUP (constr.)	-4821.9	-4821.9	10.0	10.0±0.00	N/A	N/A	1.4	1.4±0.00
K	-4240.3	-4242.9	14.0	14.2±0.40	N/A	N/A	1.0	1.0±0.00
Prop.	-4925.7	-4925.8	15.0	15.0±0.00	N/A	N/A	1.9	1.9±0.00
Oh	-4943.2	-4945.9	1.0	1.0±0.00	N/A	N/A	2.4	2.4±0.16

Table D.25: Smmary of results from the PMC sampler, using the k-add mixture proposal, on the field test data set.

Bibliography

[1] G.E. Andrews. *The Theory of Partitions*. Number v. 2 in Computers & Typesetting. Addison-Wesley Publishing Company, Advanced Book Program, 1976.

[2] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[3] Charles E. Antoniak. Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

[4] D. Avitzour. Stochastic simulation Bayesian approach to multitarget tracking. *Radar, Sonar and Navigation, IEE Proceedings* -, 142(2):41–44, Apr 1995.

[5] Y. Bar-Shalom, K.C. Chang, and H. A P Blom. Automatic track formation in clutter with a recursive algorithm. In *Proceedings of the 28th IEEE Conference on Decision and Control*, pages 1402–1408 vol.2, Dec 1989.

[6] Yaakov Bar-Shalom and Xiao-Rong Li. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[7] A. Barbu and Song-Chun Zhu. Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1239–1253, Aug 2005.

[8] M. Basseville. A genetic algorithm based multi-dimensional data asso- ciation algorithm for multi-sensor multi-target tracking. *Mathematical and Computer Modelling*, 26(4):57–69, 1997.

[9] Dimitri P. Bertsekas. Auction Algorithms for Network Flow Problems: A Tutorial Introduction. *Computational Optimization and Applications*, 1:7–66, 1992.

[10] Sebu Bjorklund, Henrik Petersson, and Gustaf Hendeby. On distinguishing between human individuals in micro-Doppler signatures. In *Radar Symposium (IRS), 2013 14th International*, volume 2, pages 865–870. IEEE, 2013.

[11] Sam Blackman, Kathy Fong, Douglas E. Carroll, Justin Lancaster, and Robert Dempster. Integration of a road network into a radar ground moving target tracking (GMTT) system and its performance evaluation, 2009.

[12] Sam Blackman, Kapriel Krikorian, Robert Rosen, Catherine Durand, and Stefan Schwoegler. Application of a joint tracking and identification method to dismount targets, 2011.

[13] Samuel Blackman and Robert Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House Publishers, 1999.

[14] D. Blackwell and J. B. Macqueen. Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1:353–355, 1973.

[15] David C. Blair. Information Retrieval, 2nd ed. *Journal of the American Society for Information Science*, 30(6):374–375, 1979.

[16] Wayne R. Blanding, Peter Willett, Roy L. Streit, and Darin Dunham. Multi-frame assignment PMHT that accounts for missed detections. In *FUSION*, pages 1–8, 2007.

[17] H. A. P. Blom and E. A. Bloem. Probabilistic data association avoiding track coalescence. *IEEE Transactions on Automatic Control*, 45(2):247–259, Feb 2000.

[18] François Bourgeois and Jean-Claude Lassalle. An Extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices. *Commun. ACM*, 14(12):802–804, December 1971.

[19] Stephen P. Brooks and Gareth O. Roberts. Assessing Convergence of Markov Chain Monte Carlo Algorithms. *Statistics and Computing*, 8:319–335, 1997.

[20] René G. Burgess and Christian J. Darken. Realistic Human Path Planning Using Fluid Simulation.

[21] Walker Butler. Design considerations for intrusion detection wide area surveillance radars for perimeters and borders. In *Technologies for Homeland Security, 2008 IEEE Conference on*, pages 47–50. IEEE, 2008.

[22] T. Cacoullos and H. Papageorgiou. Multiparameter Stirling and C-Numbers: Recurrences and Applications. *The Fibonacci Quarterly*, 22(2):119, May 1984.

[23] O. Cappé, R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert. Adaptive Importance Sampling in General Mixture Classes. *ArXiv e-prints*, October 2007.

[24] Olivier Cappé, Arnaud Guillin, Jean-Michel Marin, and Christian P. Robert. Population Monte Carlo. *JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS*, 13:907–929, 2004.

[25] George Casella. An Introduction to Empirical Bayes Data Analysis. *The American Statistician*, 39(2):83–87, 1985.

[26] George Casella, Elías Moreno, and F. Javier Girón. Cluster Analysis, Model Selection, and Prior Distributions on Models. Unpublished Technical Report, July 20111.

[27] George Casella and Christian P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, March 1996.

[28] S. Chang, R Sharan, M. Wolf, N. Mitsumoto, and J. W. Burdick. UWB radar-based tracking of humans. *Proc. IEEE Radar Con, Pasadena, CA*, 2009.

[29] SangHyun Chang, Michael Wolf, and Joel W Burdick. Human detection and tracking via ultra-wideband (UWB) radar. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 452–457. IEEE, 2010.

[30] Victor C. Chen, Fayin Li, Shen-Shyang Ho, and Harry Wechsler. Micro-Doppler effect in radar: phenomenon, model, and simulation study. *Aerospace and Electronic Systems, IEEE Transactions on*, 42(1):2–21, 2006.

[31] Chee-Yee Chong, D. Garren, and T.P. Grayson. Ground target tracking-a historical perspective. In *IEEE Aerospace Conference Proceedings*, volume 3, pages 433–448, 2000.

[32] D.E. Clark and J. Bell. Data Association for the PHD Filter. In *Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005. Proceedings of the 2005 International Conference on*, pages 217–222, Dec 2005.

[33] CNN Wire Staff. Homeland Security chief cancels costly virtual border fence. *CNN*, January 14, 2011.

[34] I. Cohen and G. Medioni. Detecting and tracking moving objects for video surveillance. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 325, 1999.

[35] S. Cong, L. Hong, and D. Wicker. Markov-chain Monte-Carlo approach for association probability evaluation. *IEE Proceedings - Control Theory and Applications*, 151(2):185–193, March 2004.

[36] Mary Kathryn Cowles and Bradley P. Carlin. Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review. *Journal of the American Statistical Association*, 91:883–904, 1996.

[37] Ingemar J. Cox and Sunita L. Hingorani. An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(2):138–150, 1996.

[38] David Frederic Crouse, Marco Guerriero, and Peter Willett. A Critical Look at the PMHT. *J. Adv. Inf. Fusion*, 4(2):93–116, 2009.

[39] G.R. Curry. *Radar System Performance Modeling*. Number v. 2 in Artech House radar library. Artech House, 2005.

[40] George C. Demos, Roberto A. Ribas, Ted J. Broida, and Samuel S. Blackman. Applications of MHT to dim moving targets, 1990.

[41] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):pp. 1–38, 1977.

[42] Benjamin W. Dilsaver. Experiments with GMTI Radar Using Micro-Doppler. Master's thesis, Brigham Young University, 2013.

[43] Armin W. Doerry. Performance Limits for Exo-Clutter Ground Moving Target Indicator (GMTI). Technical Report SAND2010-5844, Sandia National Laboratories, September 2010.

[44] R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert. Convergence of adaptive sampling schemes. Technical report, University Paris Dauphine, 2005-6.

[45] R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert. Convergence of adaptive mixtures of importance sampling schemes. *Ann. Statist.*, 35(1):420–448, 02 2007.

[46] Arnaud Doucet, Nando De Freitas, and Neil Gordon, editors. *Sequential Monte Carlo methods in practice*. 2001.

[47] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, July 2000.

[48] Oliver E. Drummond. On Features and Attributes in Multisensor, Multitarget Tracking. In *Proc. of FUSION 1999: The 2nd International Conference on Information Fusion*, July 1999.

[49] Oliver E. Drummond and David Dana-Bashian. Comparison of tracklet filter methods for non-maneuvering targets. In *Optics & Photonics 2005*, pages 59131B–59131B. International Society for Optics and Photonics, 2005.

[50] B. Efron and R. Tibshirani. An Introduction to the Bootstrap, 1993.

[51] W. J. Ewens. The sampling theory of selectively neutral alleles. *Theor. Popul. Biol.*, 3(1):87–112, March 1972.

[52] A. Farina, L. Ferranti, and G. Golino. Constrained tracking filters for A-SMGCS. In *Information Fusion, 2003. Proceedings of the Sixth International Conference of*, volume 1, pages 414–421, July 2003.

[53] Thomas S. Ferguson. A Bayesian Analysis of Some Nonparametric Problems. *Ann. Statist.*, 1(2):209–230, 03 1973.

[54] James L. Fisher and David P. Casasent. Fast JPDA multitarget tracking algorithm. *Appl. Opt.*, 28(2):371–376, Jan 1989.

[55] Robert J. Fitzgerald. Development of Practical PDA Logic for Multitarget Tracking by Microprocessor. In *American Control Conference, 1986*, pages 889–898, June 1986.

[56] David J. Fleet. Motion Models for People Tracking. In *Visual Analysis of Humans*, pages 171–198. 2011.

[57] Jr. Forney, G.D. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, March 1973.

[58] Thomas E. Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Sonar Tracking of Multiple Tartgets Using Joint Probabilistic Data Association. *IEEE Journal of Oceanic Engineering*, OE-8(3), 1983.

[59] E.B. Fox, E.B. Sudderth, D.S. Choi, and A.S. Willsky. Tracking a Non-cooperative Maneuvering Target using Hierarchical Dirichlet Processes. In *Adaptive Sensor Array Processing Conference*, Lexington, MA, June 2007.

[60] Emily B. Fox, David S. Choi, and Alan S. Willsky. Nonparametric Bayesian Methods for Large Scale Multi-Target Tracking. In *Proc. Asilomar Conf. Signals, Systems, and Computers*. IEEE, November 2006.

[61] Weina Ge and Robert T. Collins. Multi-target Data Association by Tracklets with Unsupervised Parameter Estimation. In *BMVC*, volume 2, page 5, 2008.

[62] Jonathan L Geisheimer, Eugene F Greneker III, and William S Marshall. High-resolution Doppler model of the human gait. In *AeroSense 2002*, pages 8–18. International Society for Optics and Photonics, 2002.

[63] Stuart Geman and Donald Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, November 1984.

[64] A. Genovesio and J.-C. Olivo-Marin. Split and merge data association filter for dense multi-target tracking. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, volume 4, pages 677–680 Vol.4, Aug 2004.

[65] J. L. Gertz. Multisensor Surveillance for Improved Aircraft Tracking. *MIT Lincoln Laboratory Journal*, 2(3), 1989.

[66] Charles J. Geyer. *Introduction to Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2015/03/04 2011.

[67] D. T. Gillespie. The mathematics of Brownian motion and Johnson noise. *American Journal of Physics*, 64:225–240, March 1996.

[68] Globe Newswire. Northrop Grumman Successfully Demonstrates VADER Dismount Detection, February 2010.

[69] I.R. Goodman, R.P. Mahler, and H.T. Nguyen. *Mathematics of Data Fusion*. Theory and Decision Library B. Springer Netherlands, 2013.

[70] N. Gordon. A hybrid bootstrap filter for target tracking in clutter. *Aerospace and Electronic Systems, IEEE Transactions on*, 33(1):353–358, Jan 1997.

[71] N.J. Gordon, D.J. Salmond, and A. F M Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, Apr 1993.

[72] J. Goutsias, R.P.S. Mahler, and H.T. Nguyen. *Random Sets: Theory and Applications*. The IMA Volumes in Mathematics and its Applications. Springer New York, 1997.

[73] Peter J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

[74] Sevgi Z. Gürbüz, William L. Melvin, and Douglas B. Williams. A Nonlinear-Phase Model-Based Human Detector for Radar. *Aerospace and Electronic Systems, IEEE Transactions on*, 47(4):2502–2513, 2011.

[75] Sevgi Z Gürbüz, William L Melvin, and Douglas B Williams. Kinematic Model-Based Human Detectors for Multi-Channel Radar. *Aerospace and Electronic Systems, IEEE Transactions on*, 48(2):1306–1318, 2012.

[76] F. Gustafsson and A.J. Isaksson. Best choice of coordinate system for tracking coordinated turns. In *Decision and Control, 1996., Proceedings of the 35th IEEE*, volume 3, pages 3145 –3150 vol.3, December 1996.

[77] Dennis Hancock. Radar 3D Ray Tracing, 2010.

[78] David J. Hand. Measuring Classifier Performance: A Coherent Alternative to the Area Under the ROC Curve. *Mach. Learn.*, 77(1):103–123, October 2009.

[79] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, April 1982.

[80] J.A. Hartigan. Partition models. *Communications in Statistics - Theory and Methods*, 19(8):2745–2756, 1990.

[81] Jouni Hartikainen and Simo Särkkä. *Optimal filtering with Kalman filters and smoothers – a Manual for Matlab toolbox EKF/UKF.* P.O.Box 9203, FIN-02015 TKK, Espoo, Finland, February 2008.

[82] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

[83] Ryan K. Hersey, William L. Melvin, and Edwin Culpepper. Dismount modeling and detection from small aperture moving radar platforms. In *Radar Conference, 2008. RADAR'08. IEEE*, pages 1–6. IEEE, 2008.

[84] Jeroen D. Hol, Thomas B. Schön, and Fredrik Gustafsson. On Resampling Algorithms for Particle Filters. In *Nonlinear Statistical Signal Processing Workshop*, 2006.

[85] Sture Holm. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6(2):pp. 65–70, 1979.

[86] Chang Huang, Bo Wu, and Ramakant Nevatia. Robust object tracking by hierarchical association of detection responses. In *Computer Vision–ECCV 2008*, pages 788–801. Springer, 2008.

[87] C. Hue, J.-P. Le Cadre, and P. Pérez. Tracking multiple objects with particle filtering. Technical Report 1361, IRISA, October 2000.

[88] A. Iacobucci, J.-M. Marin, and C. Robert. On variance stabilisation by double Rao-Blackwellisation. *ArXiv e-prints*, February 2008.

[89] Shane T. Jensen and Jun S. Liu. Bayesian Clustering of Transcription Factor Binding Motifs. *Journal of the American Statistical Association*, 103:188–200, 2008.

[90] Hao Jiang, S. Graillat, and R. Barrio. Accurate and Fast Evaluation of Elementary Symmetric Functions. In *Computer Arithmetic (ARITH), 2013 21st IEEE Symposium on*, pages 183–190, April 2013.

[91] S. Julier and J. K. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford, 1996.

[92] S. J. Julier and J. K. Uhlmann. A New Extension of the Kalman Filter to Nonlinear Systems. In *The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing*, 1997.

[93] Simon J. Julier. Skewed approach to filtering, 1998.

[94] Simon J. Julier and Jeffrey K. Uhlmann. A Consistent, Debiased Method for Converting Between Polar and Cartesian Coordinate Systems. In *In The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, pages 110–121, 1997.

[95] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(1):35–45, 1960.

[96] Rickard Karlsson and Fredrik Gustafsson. Monte Carlo Data Association for Multiple Target Tracking. In *Proceedings of the 2001 IEE International Seminar on Target Tracking: Algorithms and Applications*, pages 13/1–13/5, 2001.

[97] K. Kastella and C. Kreucher. Multiple model nonlinear filtering for low signal ground target applications. *Aerospace and Electronic Systems, IEEE Transactions on*, 41(2):549–564, April 2005.

[98] Robert Kaucic, Glen Brooksby, John Kaufhold, Anthony Hoogs, et al. A unified framework for tracking through occlusions and across sensor gaps. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 990–997. IEEE, 2005.

[99] Zia Khan, T. Balch, and F. Dellaert. Multitarget tracking with split and merged measurements. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 605–610 vol. 1, June 2005.

[100] Zia Khan, Tucker Balch, and Frank Dellaert. An MCMC-Based Particle Filter for Tracking Multiple Interacting Targets. In Tomás Pajdla and Jiří Matas, editors, *Computer Vision - ECCV 2004*, volume 3024 of *Lecture Notes in Computer Science*, pages 279–290. Springer Berlin Heidelberg, 2004.

[101] J. F. C. Kingman. The Representation of partition structures. *Journal of the London Math Society*, 18(2):374–380, 1978.

[102] T. Kirubarajan and Y. Bar-Shalom. Tracking evasive move-stop-move targets with a GMTI radar using a VS-IMM estimator. *IEEE Transactions on Aerospace and Electronic Systems*, 39(3):1098–1103, July 2003.

[103] T. Kirubarajan, Y. Bar-shalom, K.R. Pattipati, and I Kadar. Ground target tracking with variable structure IMM estimator. *IEEE Transactions on Aerospace and Electronic Systems*, 36(1):26–46, Jan 2000.

[104] Juho Kokkala and Simo Särkkä. Combining Particle MCMC with Rao-Blackwellized Monte Carlo Data Association for Parameter Estimation in Multiple Target Tracking, September 2014.

[105] Jost Koller and Martin Ulmke. Data fusion for ground moving target tracking. *Aerospace Science and Technology*, 11(4):261 – 270, 2007. {COGIS} '06.

[106] Y. Koren and J. Borenstein. Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation. In *Proceedings of the IEEE Conf. on Robotics and Automation*, pages 1398–1404.

[107] James T. Kost and Michael P. McDermott. Combining dependent P-values. *Statistics & Probability Letters*, 60(2):183–190, 2002.

[108] E. Kulinskaya. On two-sided p-values for non-symmetric distributions. *ArXiv e-prints*, October 2008.

[109] T. Kurien. Issues in the design of practical multitarget tracking algorithms. In Yaakov Bar-Shalom, editor, *Multitarget-multisensor tracking : advanced applications*, Artech House radar library, pages 43–83. Artech house, Norwood, MA, London, 1990.

[110] H. J. Kushner. Dynamical Equations for Optimum Non-linear Filtering. *Journal of Differential Equations*, 3:179–190, 1967.

[111] Jaeyong Lee, Fernando A. Quintana, Peter Müller, and Lorenzo Trippa. Defining Predictive Probability Functions for Species Sampling Models. Technical report, November 2008.

[112] Uri N. Lerner. *Hybrid Bayesian Networks for Reasoning about Complex Systems*. PhD thesis, Stanford University, October 2002.

[113] D. Lerro and Y. Bar-Shalom. Tracking with debiased consistent converted measurements versus EKF. *Aerospace and Electronic Systems, IEEE Transactions on*, 29(3):1015–1022, Jul 1993.

[114] Shaoyu Li, Barry L. Williams, and Yuehua Cui. A combined p-value approach to infer pathway regulations in eQTL mapping. 4:389–401+, 2011.

[115] Xiao-Rong Li and Y. Bar-Shalom. Multiple-model estimation with variable structure. *Automatic Control, IEEE Transactions on*, 41(4):478–493, Apr 1996.

[116] Lin Lin, Yaakov Bar-Shalom, and Thiagalingam Kirubarajan. Data association combined with the probability hypothesis density filter for multitarget tracking, 2004.

[117] William A. Link and Mitchell J. Eaton. On thinning of chains in MCMC. *Methods in Ecology and Evolution*, 3(1):112–115, 2012.

[118] C. Louste and A. Liegeois. Robot path planning using models of fluid mechanics. In S.G. Tzafestas and G. Schmidt, editors, *Progress in system and robot analysis and control design*, volume 243 of *Lecture Notes in Control and Information Sciences*, pages 515–524. Springer London, 1999.

[119] T. Luginbuhl, Y. Sun, and P. Willett. A track management system for the PMHT algorithm. In *Proceedings of the 4th International Conference on Information Fusion*, Montreal, Canada, 2001.

[120] Yunqian Ma, Qian Yu, and Isaac Cohen. Multiple Hypothesis Target Tracking Using Merge and Split of Graph's Nodes. In George Bebis, Richard Boyle, Bahram Parvin, et al., editors, *Advances in Visual Computing*, volume 4291 of *Lecture Notes in Computer Science*, pages 783–792. Springer Berlin Heidelberg, 2006.

[121] R. Mahler. PHD filters of higher order in target number. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4):1523–1543, October 2007.

[122] R. P. S. Mahler. The Random Set Approach to Data Fusion. In *Proceedings of the SPIE*, volume 2234, pages 287–295. Sadjadi, F.A., 1994.

[123] Ronald Mahler. A theory of PHD filters of higher order in target number, 2006.

[124] R.P.S. Mahler. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, Oct 2003.

[125] WilliamA. Massey, GeraldineA. Parker, and Ward Whitt. Estimating the parameters of a nonhomogeneous Poisson process with linear rate. *Telecommunication Systems*, 5(2):361–388, 1996.

[126] Thomas P. Minka. From Hidden Markov Models to Linear Dynamical Systems. Technical report, Tech. Rep. 531, Vision and Modeling Group of Media Lab, MIT, 1999.

[127] C. Morefield. Application of 0-1 integer programming to multitarget tracking problems. *Automatic Control, IEEE Transactions on*, 22(3):302–312, Jun 1977.

[128] J. Munkres. Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[129] Katta G. Murty. An Algorithm for Ranking all the Assignments in Order of Increasing Cost. *Operations Research*, 16(3):682–687, 1968.

[130] D. Musicki, R. Evans, and S. Stankovic. Integrated probabilistic data association. *IEEE Transactions on Automatic Control*, 39(6):1237–1241, Jun 1994.

[131] Radford M. Neal. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

[132] Andrew Y. Ng and Michael I. Jordan. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems 14 (NIPS 2001)*, pages 841–848. MIT Press, 2001.

[133] O. Nichtern and S.R. Rotman. Tracking of a Point Target in an IR Sequence using Dynamic Programming Approach. In *IEEE 24th Convention of Electrical and Electronics Engineers in Israel*, pages 265–269, Nov 2006.

[134] P.O. Nouguès and D.E. Brown. We Know Where You Are Going: Tracking Object in Terrain. *Journal of Mathematics Applied in Business and Industry*, 8:39–58, 1997.

[135] Songhwai Oh, Stuart Russell, and Shankar Sastry. Markov Chain Monte Carlo Data Association for Multiple-Target Tracking. Technical Report UCB//ERL M05/19, University of California, Berkeley, June 2005.

[136] Umut Orguner, Thomas Schön, and Fredrik Gustafsson. Improved Target Tracking with Road Network Information. Technical Report LiTH-ISY-R-2873, Linköpings universitet, January 2009.

[137] B. Pannetier, V. Nimier, and M. Rombaut. Multiple ground target tracking. *Aerospace Science and Technology*, 11(4):271 – 278, 2007. {COGIS} '06.

[138] K. Panta, Ba-Ngu Vo, and Sumeetpal Singh. Novel data association schemes for the probability hypothesis density filter. *IEEE Transactions on Aerospace and Electronic Systems*, 43(2):556–570, April 2007.

[139] Kusha Panta, Daniel E. Clark, and Ba-Ngu Vo. Data association and track management for the Gaussian mixture probability hypothesis density filter. *Aerospace and Electronic Systems, IEEE Transactions on*, 45(3):1003–1016, 2009.

[140] Panos M. Pardalos and Leonidas S. Pitsoulis. *Nonlinear assignment problems: algorithms and applications*. Springer, 2000.

[141] Hanna Pasula, Stuart J. Russell, Michael Ostland, and Yaacov Ritov. Tracking Many Objects with Many Sensors. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1160–1171, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[142] A.G.A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and Wensheng Hu. Multi-Object Tracking Through Simultaneous Long Occlusions and Split-Merge Conditions. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 666–673, June 2006.

[143] Jim Pitman. Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, 102(2):145–158, 1995.

[144] Jim Pitman. *Some developments of the Blackwell-MacQueen urn scheme*, volume Volume 30 of *Lecture Notes–Monograph Series*, pages 245–267. Institute of Mathematical Statistics, Hayward, CA, 1996.

[145] Jim Pitman and Marc Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Ann. Probab.*, 25(2):855–900, 04 1997.

[146] Aubrey B. Poore. Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking. *Comput. Optim. Appl.*, 3(1):27–57, 1994.

[147] Aubrey B. Poore and Alexander J. Robertson, III. A New Lagrangian Relaxation Based Algorithm for a Class ofMultidimensional Assignment Problems. *Comput. Optim. Appl.*, 8(2):129–150, September 1997.

[148] Aubrey P. Poore and Nenad Rijavec. A Lagrangian Relaxation Algorithm for Multidimensional Assignment Problems Arising from Multitarget Tracking. *SIAM J. Optim.*, 3(3):544–563, 1993.

[149] David Martin Ward Powers. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *International Journal of Machine Learning Technology*, 2(1):37–63, 2011.

[150] Julia Preston. Homeland Security Cancels 'Virtual Fence' After $1 Billion Is Spent. *The New York Times*, page A11, January 15, 2011.

[151] J. Prokaj, M. Duchaineau, and G. Medioni. Inferring tracklets for multi-object tracking. pages 37–44, 2011.

[152] James Gary Propp and David Bruce Wilson. Exact Sampling with Coupled Markov Chains and Applications to Statistical Mechanics. In *Proceedings of the Seventh International Conference on Random Structures and Algorithms*, pages 223–252, New York, NY, USA, 1996. John Wiley & Sons, Inc.

[153] Graham Pulford and Barbara La Scala. Over-The-Horizon Radar Tracking Using The Viterbi Algorithm - Third Report to High Frequency Radar Division. Technical Report 27/95, Adelaide, Australia, December 1995.

[154] G.W. Pulford and B.F. La Scala. Multihypothesis Viterbi Data Association: Algorithm Development and Assessment. *IEEE Transactions on Aerospace and Electronic Systems*, 46(2):583–609, April 2010.

[155] Z. S. Qin, L. A. Mccue, W. Thompson, et al. Identification of co-regulated genes through Bayesian clustering of predicted regulatory binding sites. *Nature Biotechnology*, 21(4):435–439, April 2003.

[156] T. Quach and M. Farooq. Maximum Likelihood Track Formation with the Viterbi Algorithm. In *Proc. of the 33rd IEEE Conf. on Decision and Control*, volume 1, pages 14–16, 1994.

[157] Fernando A. Quintana and Pilar L. Iglesias. Bayesian clustering and product partition models. *Journal Of The Royal Statistical Society Series B*, 65(2):557–574, 2003.

[158] Deva Ramanan, David A. Forsyth, and Andrew Zisserman. Tracking People by Learning Their Appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):65–81, 2007.

[159] H. Rauch, F. Tung, and C. Striebel. Maximum Likelihood Estimates of Linear Dynamic Systems. *AIAA Journal*, 3(8):1445–1450, 1965.

[160] D.B. Reid. An Algorithm for Tracking Multiple Targets. *IEEE Trans. Automatic Control*, AC-24:843–854, 1979.

[161] Vladimir Reilly, Haroon Idrees, and Mubarak Shah. Detection and Tracking of Large Number of Targets in Wide Area Surveillance. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, volume 6313 of *Lecture Notes in Computer Science*, pages 186–199. Springer Berlin Heidelberg, 2010.

[162] J. Richards. GMTI Radar minimum detectable velocity. *SAND2011-1767*, 2011.

[163] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[164] J.A Roecker. A class of near optimal JPDA algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 30(2):504–510, Apr 1994.

[165] J.A. Roecker. Multiple scan joint probabilistic data association. *Aerospace and Electronic Systems, IEEE Transactions on*, 31(3):1204–1210, Jul 1995.

[166] J.A Roecker and G.L. Phillis. Suboptimal joint probabilistic data association. *IEEE Transactions on Aerospace and Electronic Systems*, 29(2):510–517, Apr 1993.

[167] Jeffrey N. Rouder, Paul L. Speckman, Dongchu Sun, Richard D. Morey, and Geoffrey Iverson. Bayesian t-Tests for Accepting and Rejecting the Null Hypothesis. *Psychonomic Bulletin & Review*, 16:225–237, 2009.

[168] Yanhua Ruan and P. Willett. The turbo PMHT. *IEEE Transactions on Aerospace and Electronic Systems*, 40(4):1388–1398, Oct 2004.

[169] S. Särkkä, T. Tamminen, A. Vehtari, and J. Lampinen. Probabilistic Methods in Multiple Target Tracking: Review and Bibliography. Technical report, Helsinki University of Technology, Helsinki University of Technology, 2004.

[170] Simo Särkkä, Aki Vehtari, and Jouko Lampinen. Rao-Blackwellized Monte Carlo Data Association for multiple target tracking. In *Proc. of FUSION 2004: The 7th International Conference on Information Fusion*, June 2004.

[171] Simo Särkkä, Aki Vehtari, and Jouko Lampinen. Rao-Blackwellized particle filter for multiple target tracking. *Inf. Fusion*, 8(1):2–15, 2007.

[172] V. Savani and A. A. Zhigljavsky. Efficient Estimation of Parameters of the Negative Binomial Distribution. In *Communications in Statistics: Theory and Methods*, volume 35, pages 767–783, 2006.

[173] Y. Schutz and R. Herren. Assessment of speed of human locomotion using a differential satellite global positioning system. *Medicine and Science in Sports and Exercise*, 32(3):642–646, March 2000.

[174] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[175] Bar Y. Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica 11*, pages 451–460, 1975.

[176] Peter J. Shea, Tim Zadra, Dale M. Klamer, Ellen Frangione, and Rebecca Brouillard. Improved state estimation through use of roads in ground tracking, 2000.

[177] Hedvig Sidenbladh. Multi-target particle filtering for the probability hypothesis density. In *Proceedings of the Sixth International Conference of Information Fusion*, volume 2, pages 800–806, July 2003.

[178] Vivek Kumar Singh, Bo Wu, and Ramakant Nevatia. Pedestrian tracking by associating tracklets using detection residuals. In *Motion and video Computing, 2008. WMVC 2008. IEEE Workshop on*, pages 1–8. IEEE, 2008.

[179] Lawrence D. Stone, Carl A. Barlow, and Thomas L. Corwin. *Bayesian multiple target tracking*. Artech House, 1999.

[180] Roy L. Streit. Maximum Likelihood Method for Probabilistic Multihypothesis Tracking. In Oliver E. Drummond, editor, *Signal and Data Processing for Small Targets*, pages 394–405, 1994.

[181] Roy L. Streit and Tod E. Luginbuhl. Probabilistic Multi-Hypothesis Tracking. Technical report, NAVAL UNDERWATER SYSTEMS CENTER NEWPORT RI, 1995.

[182] Lars Svensson and Mark Morelande. Target tracking based on estimation of sets of trajectories. In *Information Fusion (FUSION), 2014 17th International Conference on*, pages 1–8. IEEE, 2014.

[183] D. Tahmoush, J. Clark, and J. Silvious. Tracking of dismounts moving in cross-range using GMTI radar. In *Radar Symposium (IRS), 2010 11th International*, pages 1–6, June 2010.

[184] D. Tahmoush and J. Silvious. Radar micro-doppler for long range front-view gait recognition. In *IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*, pages 1–6, September 2009.

[185] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[186] Z. Tu and S. C. Zhu. Image Segmentation by Data-Driven Markov Chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):657–673, 2002.

[187] Ba-Ngu Vo and Wing-Kin Ma. A closed-form solution for the probability hypothesis density filter. In *8th International Conference on Information Fusion*, volume 2, page 8 pp., July 2005.

[188] Ba-Ngu Vo and Wing-Kin Ma. The Gaussian Mixture Probability Hypothesis Density Filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, Nov 2006.

[189] Ba-Ngu Vo, Sumeetpal Singh, and A Doucet. Sequential monte carlo implementation of the phd filter for multi-target tracking. In *Proceedings of the Sixth International Conference of Information Fusion*, volume 2, pages 792–799, July 2003.

[190] Ba-Tuong Vo, Ba-Ngu Vo, and Antonio Cantoni. The Cardinalized Probability Hypothesis Density Filter for Linear Gaussian Multi-Target Models. In *40th Annual Conference on Information Sciences and Systems*, pages 681–686, March 2006.

[191] H. M. Wallach, S. T. Jensen, L. Dicker, and K. A. Heller. An Alternative Prior Process for Nonparametric Bayesian Clustering. *ArXiv e-prints*, January 2008.

[192] David Walsh. Unmanned aerostats furnish vital geospatial intelligence. *Defense Systems*, September 9, 2011.

[193] E.A. Wan and R. Van der Merwe. The unscented Kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158, 2000.

[194] Wikipedia. Nombre de Stirling — Wikipedia, The Free Encyclopedia. `http://fr.wikipedia.org/wiki/Nombre_de_Stirling`, 2011. [Online; accessed August 2011].

[195] Wikipedia. Stirling Numbers of the First Kind — Wikipedia, The Free Encyclopedia. `http://en.wikipedia.org/wiki/Stirling_numbers_of_the_first_kind`, 2011. [Online; accessed August 2011].

[196] P. Willett, Y. Ruan, and R. Streit. PMHT: Problems and Some Solutions. *IEEE Trans. on Aerospace and Electronic Systems*, 38:738–754, 2002.

[197] Fei Yan, W. Christmas, and Josef Kittler. All pairs shortest path formulation for multiple object tracking with application to tennis video analysis. In *British Machine Vision Conference*, 2007.

[198] Yiming Yang and Xin Liu. A Re-examination of Text Categorization Methods. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 42–49, New York, NY, USA, 1999. ACM.

[199] Qian Yu and G. Medioni. Integrated Detection and Tracking for Multiple Moving Objects using Data-Driven MCMC Data Association. In *Motion and video Computing, 2008. WMVC 2008. IEEE Workshop on*, pages 1–8, Jan. 2008.

[200] Qian Yu and Gérard Medioni. Map-enhanced detection and tracking from a moving platform with local and global data association. In *Motion and Video Computing, 2007. WMVC'07. IEEE Workshop on*, pages 3–3. IEEE, 2007.

[201] Qian Yu, Gérard G. Medioni, and Isaac Cohen. Multiple Target Tracking Using Spatio-Temporal Markov Chain Monte Carlo Data Association. In *CVPR*, 2007.

[202] S. L. Zabell. Predicting the unpredictable. *Synthese*, 90:205–232, 1992. 10.1007/BF00485351.

[203] Sandy L. Zabell. W. E. Johnson's 'Sufficientness' Postulate. *The Annals of Statistics*, 10(4):1090–1099, 12 1982.

[204] M.A Zaveri, S.N. Merchant, and U.B. Desai. Tracking multiple point targets using genetic interacting multiple model based algorithm. In *Proceedings of the International Symposium on Circuits and Systems (ISCAS)*, volume 3, pages III–917–20 Vol.3, May 2004.

[205] Mukesh A. Zaveri, S. N. Merchant, and Uday B. Desai. Interacting multiple-model-based tracking of multiple point targets using an expectation maximization algorithm in an infrared image sequence. *Optical Engineering*, 44(1):016402–016402–17, 2005.

[206] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.