

UC Santa Barbara

UC Santa Barbara Electronic Theses and Dissertations

Title

Numerical Level Set Methods for High-Dimensional Dynamical Systems

Permalink

<https://escholarship.org/uc/item/75v1640t>

Author

Kirchner, Matthew R.

Publication Date

2023

Peer reviewed|Thesis/dissertation

University of California
Santa Barbara

Numerical Level Set Methods for High-Dimensional Dynamical Systems

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

Doctor of Philosophy

in

Electrical and Computer Engineering

by

Matthew R. Kirchner

Committee in charge:

Professor João P. Hespanha, Chair
Professor Stanley J. Osher
Dr. Gary A. Hewer
Professor Andrew R. Teel
Professor Jason R. Marden

June 2023

The Dissertation of Matthew R. Kirchner is approved.

Professor Stanley J. Osher

Dr. Gary A. Hewer

Professor Andrew R. Teel

Professor Jason R. Marden

Professor João P. Hespanha, Committee Chair

May 2023

Numerical Level Set Methods for High-Dimensional Dynamical Systems

Copyright © 2023

by

Matthew R. Kirchner

To my wife Shawnasie, my son Kai, and my daughter Alyeska.

Acknowledgements

I wish to thank all the great mentors and collaborators that shaped my career including, but not limited to, Gary Hewer, Larry Peterson, Stan Osher, Jerome Darbon, and my advisor Joao Hespanha. I thank the Office of Naval Research for supporting a large amount of the research presented in this thesis. It was a once-in-a-lifetime learning experience that will forever shape my research career. I owe everything to my family. To my wife Shawnasie, thank you for supporting me through school, moving back and forth across the west, and being patient while I continued research on nights and weekends for all these years. To my family and friends: thank you for encouraging me and for your prayers. I especially appreciate all that visited us at various stops we made across this great country. And most importantly, I want to thank God for providing me with such an incredible opportunity.

Curriculum Vitæ

Matthew R. Kirchner

Education

- 2023 Ph.D. in Electrical and Computer Engineering (Expected), University of California, Santa Barbara.
- 2013 M.S. in Electrical Engineering, University of Colorado.
- 2007 B.S. in Mechanical Engineering, Washington State University.

Publications

Journal Articles

1. M. R. Kirchner, S. R. Kirchner, A. A. Dennis, and S. E. Rigby, “Non-parametric Characterization of Blast Loads,” to appear in *International Journal of Protective Structures*.
2. D. Grimsman, M. R. Kirchner, J. P. Hespanha, and J. R. Marden, “The Impact of Measurement Passing in Sensor Network Measurement Selection,” in *IEEE Transactions on Control of Network Systems*, volume 10, issue 1, pp. 112-123, 2023 [1].
3. M. R. Kirchner, R. Mar, G. Hewer, J. Darbon, S. Osher, and Y.T. Chow, “Time-Optimal Collaborative Guidance Using the Generalized Hopf Formula,” *IEEE Control Systems Letters*, volume 2, issue 2, pp. 201-206, 2018.

Conference Proceedings

1. M. R. Kirchner, D. Grimsman, J. P. Hespanha, and J. R. Marden, “Trajectories for the Optimal Collection of Information,” accepted to appear in the proceedings of the *IEEE Aerospace Conference*, 2023 [2].
2. D. Grimsman, M. R. Kirchner, J. P. Hespanha, and J. R. Marden, “The Impact of Message Passing in Agent-Based Submodular Maximization,” in *IEEE Conference on Decision and Control (CDC)*, pp. 530-535, 2020 [3].
3. M. R. Kirchner, M. J. DeBord, and J. P. Hespanha, “A Hamilton-Jacobi Formulation for Optimal Coordination of Heterogeneous Multiple Vehicle Systems,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11623-11630, 2020.
4. M. R. Kirchner, E. Ball, J. Hoffer, and D. Gaublomme, “Reachability as a Unifying Framework for Computing Helicopter Safe Operating Conditions and Autonomous Emergency Landing,” in *21st IFAC World Congress*, IFAC PapersOnLine, volume 53, issue 2, pp. 9282-9287, 2020.
5. M. R. Kirchner, J. P. Hespanha, and D. Garagic, “Heterogeneous Measurement Selection for Vehicle Tracking using Submodular Optimization,” in *IEEE Aerospace Conference*, pp. 1-10, 2020.

6. M. R. Kirchner, "A Level Set Approach to Online Sensing and Trajectory Optimization with Time Delays," in *10th IFAC Symposium on Intelligent Autonomous Vehicles*, IFAC PapersOnLine, volume 52, issue 8, pp. 301-306, 2019.
7. M. R. Kirchner, G. Hewer, J. Darbon, and S. Osher, "A Primal-Dual Method for Optimal Control and Trajectory Generation in High-Dimensional Systems," in *IEEE Conference on Control Technology and Applications*, pp. 1575-1582, 2018.
8. M. R. Kirchner, K. Ryan, and N. Wright, "Maneuvering Vehicle Tracking with Bayesian Change-point Detection," in *IEEE Aerospace Conference*, pp. 1-9, 2017.
9. M. R. Kirchner, "Automatic Thresholding of SIFT Descriptors," in *IEEE International Conference on Image Processing*, pp. 291-295, 2016.
10. M. R. Kirchner and S. R. Kirchner, "Approximating Warhead Service Life Using Bayesian Non-Parametric Degradation Modeling," in *43rd JANNAF Structures and Mechanical Behavior Subcommittee Meeting*, 2015.

Patents

1. M. R. Kirchner, K. Ryan, and N. Wright, "System and Method for Tracking Dynamic Maneuvering Vehicles," Navy Case Number 107010. Filed under secrecy order: 2017; Notice of allowance: 2020.
2. M. R. Kirchner, "Image-Matching Navigation Using Thresholding of Local Image Descriptors," United States Patent No. 10402682, Issued September 3, 2019.
3. M. R. Kirchner, "System and Method for Thresholding of Local Image Descriptors," United States Patent No. 10346717, Issued July 9, 2019.

Abstract

Numerical Level Set Methods for High-Dimensional Dynamical Systems

by

Matthew R. Kirchner

Presented here is research focused on numerical advancements in Hamilton-Jacobi (HJ) theory as they provide a fundamental tool to address many problems in autonomous robotics such as optimal trajectory planning, safety critical reactive control, pursuit-evasion, and the optimal gathering of information of an unknown environment. HJ equations have had limitations in the past for computing usable solutions due to poor scaling with respect to system dimension. This was because a spatial grid had to be constructed densely in each dimension. Two general frameworks are proposed to overcome this limitation. First, methods based on trajectory optimization are presented and in particular, those based on generalizations of the Hopf formula are developed. This class of methods leverage the fact that for many real-world problems, only pointwise solutions are necessary, and this enables these optimization-based approaches to be implemented on embedded hardware for real-time operation. Second, decomposition methods are developed. This class of methods leverage certain problem structures that allow smaller-dimensional subproblems to be formed. The solutions of these subproblems can then be aggregated to compute the global solution of the original HJ equation.

Contents

Curriculum Vitae	vi
Abstract	viii
1 Introduction	1
1.1 Contributions	2
Part I Trajectory Optimization Approaches to Solving Hamilton–Jacobi Equations	7
2 A Primal-Dual Method for Optimal Control and Trajectory Generation in High-Dimensional Systems	9
2.1 Introduction	9
2.2 Solutions to Hamilton–Jacobi Equations with the Hopf Formula	11
2.3 Proximal Splitting Methods for Control	15
2.4 Time-Optimal Control	20
2.5 Trajectory Generation with the Generalized Hopf Formula	22
2.6 Results	25
2.7 Conclusion	33
3 Time-Optimal Collaborative Guidance Using the Generalized Hopf Formula	34
3.1 Introduction	34
3.2 Pursuit-Evasion Model	36
3.3 Hamilton–Jacobi Equations with Bounded Control	41
3.4 Time-Optimal Control with the Hopf Formula	46
3.5 Results	50
3.6 Conclusions and Future Work	52

Part II Decomposition Approaches to Solving Hamilton–Jacobi Equations **54**

4	A Hamilton–Jacobi Formulation for Optimal Coordination of Heterogeneous Multiple Vehicle Systems	56
4.1	Introduction	56
4.2	Problem Formulation	60
4.3	Hamilton–Jacobi Equations For Optimal Coordination	62
4.4	A Level Set Method with the Generalized Hopf Formula	68
4.5	Results	73
4.6	Conclusions and Future Work	77
5	Heterogeneous Measurement Selection for Vehicle Tracking using Submodular Optimization	78
5.1	Introduction	78
5.2	Fisher Information Matrix	82
5.3	FIM-Based Measurement Selection	83
5.4	Motion Models	90
5.5	Measurement Models	92
5.6	Results	99
5.7	Conclusions and Future Work	103
6	Trajectories for the Optimal Collection of Information	105
6.1	Introduction	105
6.2	The Vehicle Sensing Problem	108
6.3	Decomposition of Coupled Systems	112
6.4	HJB Decomposition	119
6.5	Optimal Information Collection	124
6.6	Results	126
6.7	Conclusion	129
A	Appendix for Chapter 4	131
B	Appendix for Chapter 6	134
	Bibliography	137

Chapter 1

Introduction

Demand for increased capabilities in robotics has dramatically increased over the last decade, touching virtually every segment of the economy. This includes manufacturing, defense, medical equipment, transportation, space exploration, and agriculture. The demand is evidenced by growing investment in the field of robotics from both industry and government. In particular, there is a demand for heightened levels of autonomy for these application areas as their system complexity continues to grow. This will lead to large-scale, mixed systems, consisting of fully autonomous agents interacting with semi-autonomous (human-in-the-loop) and non-autonomous agents. There exist many technological challenges that must be addressed before these systems become more incorporated into society, forming the basis of this research. Among these challenges is motion planning of both single and multiple vehicles, reactive safety of vehicle systems, and how information from onboard sensors is gathered and shared among the group. Multi-vehicle motion planning is of interest as we encounter increasingly dense environments in both airspace and roadway traffic. By motion planning salvos of vehicles, we improve upon system performance and reduce the congestion that results from entirely self-interested path planning. Secondly, how information is collected and shared within a system is

of significance as real-world systems ultimately lack the infrastructure for centralized planning. Additionally, how information is collected should be studied as this would direct how unknown areas are explored and maximize the information gain from the environment. Lastly, we must consider reactive control for the inevitable, unforeseen circumstances. Reactive methods related to safety must have strong theoretical guarantees that a collision can be avoided but at the same time it's essential the solution can be computed rapidly on embedded hardware using on-board sensors. An illustration of the reactive collision avoidance problem is given in Figure 1.0.1.

Presented here is research focused on numerical advancements in Hamilton-Jacobi (HJ) theory as they provide a fundamental tool to address many problems in autonomous robotics, and can be used for single and multi-vehicle path planning and reactive collision avoidance. HJ equations have had limitations in the past for computing usable solutions due to poor scaling with respect to system dimension. This was due to the fact that a spatial grid had to be constructed densely in each dimension. This exponential dimensional scaling in optimization is sometimes referred to as the “curse of dimensionality”. Creating equivalent formulations that no longer require spatial grids leads to methods that can execute in real-time on embedded hardware and is focus of the following research. Some key contributions as listed below.

1.1 Contributions

1. Presented is a method for efficient computation of the Hamilton–Jacobi (HJ) equation for time-optimal control problems using the generalized Hopf formula. Typically, numerical methods to solve the HJ equation rely on a discrete grid of the solution space and exhibit exponential scaling with dimension. The generalized Hopf formula avoids the use of grids and numerical gradients by formulating an

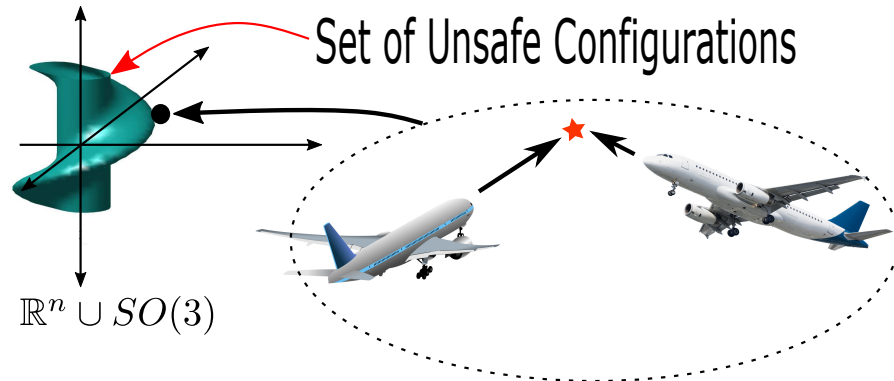


Figure 1.0.1: After detection of a previously unknown vehicle in the airspace, the relative configuration of the vehicles can be represented as a single point in a high dimensional space. We need to rapidly determine if the configuration is unsafe so the vehicle can react. The set of unsafe configurations is shown with the green shape. A configuration outside of the green shape is safe and no action is necessary. A point on or inside the green shape is unsafe and an evasive maneuver needs to be initiated. Such unsafe regions can be characterized by specific Hamilton-Jacobi equations. Rapidly computing pointwise solutions, as is the focus of this research, enables such safety critical applications to be realized.

unconstrained convex optimization problem. The solution at each point is completely independent, and allows a massively parallel implementation if solutions at multiple points are desired. This work presents a primal-dual method for efficient numeric solution and presents how the resulting optimal trajectory can be generated directly from the solution of the Hopf formula, without further optimization. Examples presented have execution times on the order of milliseconds and experiments show computation scales approximately polynomial in dimension with very small high-order coefficients.

2. Presented is a new method for calculating the time-optimal guidance control for a multiple vehicle pursuit-evasion system. A joint differential game of k pursuing vehicles relative to the evader is constructed, and a Hamilton–Jacobi–Isaacs (HJI) equation that describes the evolution of the value function is formulated. The value function is built such that the terminal cost is the squared distance from the

- boundary of the terminal surface. Additionally, all vehicles are assumed to have bounded controls. Typically, a joint state space constructed in this way would have too large a dimension to be solved with existing grid-based approaches. The value function is computed efficiently in high-dimensional space, without a discrete grid, using the generalized Hopf formula. The optimal time-to-reach is iteratively solved, and the optimal control is inferred from the gradient of the value function.
3. We present a method for optimal coordination of multiple vehicle teams when multiple endpoint configurations are equally desirable, such as seen in the autonomous assembly of formation flight. The individual vehicles' positions in the formation are not assigned a priori and a key challenge is to find the optimal configuration assignment along with the optimal control and trajectory. Commonly, assignment and trajectory planning problems are solved separately. We introduce a new multi-vehicle coordination paradigm, where the optimal goal assignment and optimal vehicle trajectories are found simultaneously from a viscosity solution of a single Hamilton–Jacobi (HJ) partial differential equation (PDE), which provides a necessary and sufficient condition for global optimality. Intrinsic in this approach is that individual vehicle dynamic models need not be the same, and therefore can be applied to heterogeneous systems. Numerical methods to solve the HJ equation have historically relied on a discrete grid of the solution space and exhibits exponential scaling with system dimension, preventing their applicability to multiple vehicle systems. By utilizing a generalization of the Hopf formula, we avoid the use of grids and present a method that exhibits polynomial scaling in the number of vehicles.
 4. We study a scenario where a group of agents, each with multiple heterogeneous sensors are collecting measurements of a vehicle and the measurements are transmitted

over a communication channel to a centralized node for processing. The communication channel presents an information-transfer bottleneck as the sensors collect measurements at a much higher rate than what is feasible to transmit over the communication channel. In order to minimize the estimation error at the centralized node, only a carefully selected subset of measurements should be transmitted. We propose to select measurements based on the Fisher information matrix (FIM), as “minimizing” the inverse of the FIM is required to achieve small estimation error. Selecting measurements based on the FIM leads to a combinatorial optimization problem. However, when the criteria used to select measurements is both monotone and submodular it allows the use of a greedy algorithm that is guaranteed to be within $1 - 1/e \approx 63\%$ of the optimum and has the critical benefit of quadratic computational complexity. To illustrate this concept, we derive the FIM criterion for different sensor types to which we apply FIM-based measurement selection. The criteria considered include the time-of-arrival and Doppler shift of passively received radio transmissions as well as detected key-points in camera images.

5. We study a scenario where an aircraft has multiple heterogeneous sensors collecting measurements to track a target vehicle of unknown location. The measurements are sampled along the flight path and our goals to optimize sensor placement to minimize estimation error. We select as a metric the Fisher Information Matrix (FIM), as “minimizing” the inverse of the FIM is required to achieve small estimation error. We propose to generate the optimal path from the Hamilton–Jacobi (HJ) partial differential equation (PDE) as it is the necessary and sufficient condition for optimality. A traditional method of lines (MOL) approach, based on a spatial grid, lends itself well to the highly non-linear and non-convex structure of the problem induced by the FIM matrix. However, the sensor placement problem

results in a state space dimension that renders a naive MOL approach intractable. We present a new hybrid approach, whereby we decompose the state space into two parts: a smaller subspace that still uses a grid and takes advantage of the robustness to non-linearities and non-convexities, and the remaining state space that can be found efficiently from a system of ODEs, avoiding formation of a spatial grid.

Part I

Trajectory Optimization Approaches to Solving Hamilton–Jacobi Equations

This section is devoted to the development of optimization techniques to compute pointwise solutions to the HJ equation. This is accomplished by constructing a trajectory optimization problem in such a fashion that the solution of which coincides with the viscosity solution of the HJ equation. This class of methods avoids a spatial grid by replacing it with a scalar temporal grid, which can be computed in an efficient manner. It is shown that solutions can be computed in mere milliseconds, demonstrating this technique is well suited for problems that need to be computed on-line in order to respond to unpredictable situations. Chapter 2 develops an optimization method, based on proximal splitting, that rapidly computes the HJ equation for optimal control problems.

An intriguing area of research that was investigated is the generalization of these methods to differential games, which will open the door to many useful applications. For instance, in the design of safety critical systems we ideally want to create methods that not only react to unsafe events as they arise, but also *preemptively* react to avoid dangerous situations before they occur. The differential game formulation, along with fast numerical methods to solve their associated HJ equation, provide a rigorous mathematical framework to solve such problems. Chapter 3 documents promising early work that showed a generalization of the Hopf formula could be utilized for certain non-convex Hamiltonians, thereby facilitating their use on game problems. The method is then validated on a multi-vehicle pursuit-evasion problem.

Chapter 2

A Primal-Dual Method for Optimal Control and Trajectory Generation in High-Dimensional Systems

2.1 Introduction

Hamilton–Jacobi equations play a fundamental role in optimal control theory as they establish sufficient conditions for optimality [4]. Traditionally, numerical solutions to HJ equations require a dense, discrete grid of the solution space [5, 6, 7]. Computing the elements of this grid scales poorly with dimension and has limited use for problems with dimension greater than four. The exponential dimensional scaling in optimization is sometimes referred to as the “curse of dimensionality” [8, 9]. Recent research [10, 11] has discovered numerical solutions based on the generalized Hopf formula that do not require a grid and can be used to efficiently compute solutions of a certain class of Hamilton–Jacobi PDEs that arise in linear control theory and differential games.

A key hurdle in the development of efficient high-dimensional solutions is the time-

dependent Hamiltonian that results for general control problems. Kirchner et al. [11] applied the generalized Hopf formula with time-dependent Hamiltonian to efficiently solve multi-vehicle optimal pursuit-evasion using the linearized models found in [12]. In that work, the unique, specific structure of the model was used to derive a closed form solution to the gradient of the objection function, thereby allowing efficient optimization. This same technique cannot be applied to general linear systems and attempts to use numeric gradient approximation would increase computation time.

Darbon and Osher presented a proximal splitting algorithm in [10] using the split Bregman/ADMM approach [13], but this only applies to systems with time-independent Hamiltonians of the form $\dot{x} = f(u(t))$, and has limited use for general linear control problems. Chow et al. [14] developed a coordinate descent method, but this optimization method lacks robustness for the nonsmooth optimization that typically result from the Hopf formula for optimal control problems.

This work presents a parallel proximal splitting optimization method [15] for solving time-optimal control problems with the generalized Hopf formula, including those with time-dependent Hamiltonians. This allows efficient solutions to generalized linear models, even when no explicit gradient of the objective function is known and without resorting to consensus-type algorithms [16]. Section 2.2 reviews using the Hopf formula for solutions to the Hamilton–Jacobi equations that arise in optimal linear control and largely follows the work of [11] and [10]. The main contributions of this paper are presented in Section 2.3, with a primal-dual method for solving the Hopf formula, Section 2.4, which presents obtaining the optimal control, and Section 2.5, where the optimal trajectory can be obtained directly from the solution of the Hopf formula. The new methods are applied on various time-optimal control problems and are presented in Section 2.6.

2.2 Solutions to Hamilton–Jacobi Equations with the Hopf Formula

Consider system dynamics represented as

$$\dot{x}(t) = f(u(t)), \quad (2.2.1)$$

where $x(t) \in \mathbb{R}^n$ is the system state and $u(t) \in \mathcal{U} \subset \mathbb{R}^m$ is the control input, constrained to lie in the convex admissible control set \mathcal{U} . The system in (2.2.1) describes how the state evolves in time and is considered a dynamic constraint when control inputs $u(t)$ are to be optimized. We consider a cost functional for a given initial time t , and terminal time T

$$K(x, u, t) = \int_t^T L(u(s)) ds + J(x(T)), \quad (2.2.2)$$

where $x(T)$ is the solution of (2.2.1) at terminal time, T . We assume that the terminal cost function $J : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex. The function $L : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$ is the running cost, and represents the rate that cost is accrued. The value function $v : \mathbb{R}^n \times (-\infty, T] \rightarrow \mathbb{R}$ is defined as the minimum cost, K , among all admissible controls for a given state x , and time $t \leq T$ with

$$v(x, t) = \inf_{u \in \mathcal{U}} K(x, u, t). \quad (2.2.3)$$

The value function in (2.2.3) satisfies the dynamic programming principle [17, 18] and also satisfies the following initial value Hamilton–Jacobi (HJ) equation by defining the function $\varphi : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$ as $\varphi(x, t) = v(x, T - t)$, with φ being the viscosity solution of

$$\begin{cases} \frac{\partial \varphi}{\partial t}(x, t) + H(\nabla_x \varphi(x, t)) = 0 & \text{in } \mathbb{R}^n \times (0, +\infty), \\ \varphi(x, 0) = J(x) & \forall x \in \mathbb{R}^n, \end{cases} \quad (2.2.4)$$

where the Hamiltonian $H : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is defined by

$$H(p) = \sup_{c \in \mathbb{R}^m} \{\langle -f(c), p \rangle - L(c)\}. \quad (2.2.5)$$

We proceed with the Hamilton Jacobi formulation for time-optimal control to reach some convex terminal set Ω , though the following methods can be generalized to other optimization problems. To apply the constraint that the control must be bounded, we introduce the following running cost $L = \mathcal{I}_{\mathcal{U}}$, where $\mathcal{I}_{\mathcal{C}} : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is the indicator function for the set \mathcal{C} and is defined by

$$\mathcal{I}_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C} \\ +\infty & \text{otherwise.} \end{cases}$$

This reduces the Hamiltonian to

$$H(p) = \max_{c \in \mathcal{U}} \langle -f(c), p \rangle.$$

Solving the HJ equation (2.2.4) describes how the value function evolves with time at any point in the state space, and from this optimal control policies can be found.

2.2.1 Viscosity Solutions with the Hopf Formula

It was shown in [18] that an exact, point-wise viscosity solution to (2.2.4) can be found using the Hopf formula [19]. Moreover, no discrete grid is constructed, and the formula can provide a numerical method that is efficient even when the state space is

high-dimensional. The value function can be found with the Hopf formula

$$\varphi(x, t) = -\min_{p \in \mathbb{R}^n} \{J^*(p) + tH(p) - \langle x, p \rangle\}, \quad (2.2.6)$$

where the Fenchel–Legendre transform $g^* : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ of a convex, proper, lower semicontinuous function $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is defined by [20]

$$g^*(p) = \sup_{x \in \mathbb{R}^n} \{\langle p, x \rangle - g(x)\}. \quad (2.2.7)$$

2.2.2 General Linear Models

Now consider the following linear state space model

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (2.2.8)$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, state vector $x \in \mathbb{R}^n$, and control input $u \in \mathcal{U} \subset \mathbb{R}^m$. We can make a change of variables

$$z(t) = e^{-tA}x(t), \quad (2.2.9)$$

which results in the following system

$$\dot{z}(t) = e^{-tA}Bu(t), \quad (2.2.10)$$

with terminal cost function now defined in z with

$$\varphi(z, 0) = J_z(z, 0) = J_x(e^{TA}z). \quad (2.2.11)$$

For clarity in the sections to follow, we use the notation H_z to refer to the Hamiltonian for systems defined by (2.2.10), and H_x for systems defined by (2.2.8). Additionally, with a slight abuse of notation, we denote by $J_x^*(p, T)$ the Fenchel transform of $J(x, t)$ with respect to the variable x at time $t = T$. Notice that the system (2.2.10) is now time-varying, and it was shown in [21, Section 5.3.2, p. 215] that the Hopf formula can be generalized for a time-dependent Hamiltonian to solve for the value function of the system in (2.2.10) with

$$\begin{aligned} \varphi(z_0, T) = & -\min_{p \in \mathbb{R}^n} \left\{ J_z^*(p, 0) \right. \\ & \left. + \int_0^T H_z(p, s) ds - \langle z_0, p \rangle \right\}, \end{aligned} \tag{2.2.12}$$

with H_z defined as

$$H_z(p, t) = \max_{c \in \mathcal{U}} \langle -e^{-(T-t)A} Bc, p \rangle. \tag{2.2.13}$$

The change of variable to $(T - t)$ is required for time since the problem was converted to an initial value formulation from a terminal value formulation in (2.2.4). The value function found by solving the unconstrained optimization problem in (2.2.12) can be thought of as the minimal cost of a system starting at initial state $z(0) = z_0$ and ending at terminal state $z(T)$.

Remark 2.1. While we are solving an initial value problem in this work, we can solve for a candidate solution of a two point boundary value problem (TPBVP) by selecting for the terminal set Ω a ball, and shrinking the radius until we get arbitrarily close to the terminal boundary condition of the corresponding TPBVP [22, Section 2.7.2, p. 66].

2.3 Proximal Splitting Methods for Control

Proximal splitting methods [15] are a powerful group of convex optimization algorithms that efficiently solve non-smooth minimization problems in high dimensions. These methods are used for problems of the form

$$\min_{p \in \mathbb{R}^n} G(p) + F(p),$$

where proximal points for G and F can be easily computed. These family of methods have been proven effective for image processing and compressed sensing applications [13]. The proximal point of f at p for some $\alpha > 0$ is given by

$$(I + \alpha \partial f)^{-1}(p) = \arg \min_{w \in \mathbb{R}^n} \left\{ \alpha f(w) + \frac{1}{2} \|w - p\|_2^2 \right\}.$$

The primal-dual algorithm [23] is a proximal splitting algorithm that solves the minimization problem of the form

$$\min_{p \in \mathbb{R}^n} G(p) + F(Kp), \tag{2.3.1}$$

with $K \in \mathbb{R}^{m \times n}$ and G, F being assumed convex, by converting (2.3.1) to the saddle-point problem

$$\min_{p \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m} \langle Kp, y \rangle + G(p) - F^*(y).$$

The minimizer can be found by iterating the following update procedure

$$\begin{cases} y^{k+1} &= (I + \sigma \partial F^*)^{-1} (y^k + \sigma K \bar{p}^k) \\ p^{k+1} &= (I + \tau \partial G)^{-1} (p^k - \tau K^\top y^{k+1}) \\ \bar{p}^{k+1} &= p^{k+1} + \theta (p^{k+1} - p^k), \end{cases} \quad (2.3.2)$$

until convergence with $\tau, \sigma > 0$ being the primal and dual step sizes and $\theta \in [0, 1]$. It was shown in [23] that the primal-dual method of (2.3.2) converges at a rate of $\mathcal{O}(1/k)$ for general convex functions F and G provided the condition $\tau\sigma \|K\|^2 < 1$ is satisfied. If one or both of F and G are strongly convex, then [23] provides an alteration to the algorithm in (2.3.2) that was shown to converge at a faster rate. While some problems in Section 2.6 meet this criteria, these accelerated algorithms are outside the scope of this work, and all examples used the algorithm in (2.3.2).

2.3.1 Primal-Dual Solutions to the Generalized Hopf Formula

Suppose \mathcal{U} is a closed convex set such that $0 \in \text{int}\mathcal{U}$, where $\text{int}\mathcal{U}$ denotes the interior of the set \mathcal{U} . Then $(\mathcal{I}_{\mathcal{U}})^*$ defines a norm $\|(\cdot)\|$ and we denote by $\|(\cdot)\|_*$ its dual norm [20]. Also consider that the set \mathcal{U} can be scaled by an injective linear transformation, Q^{-1} , to give the appropriate problem-specific control bound, then (2.2.13) can be written as

$$H_z(p, t) = \left\| \left(-e^{-(T-t)A} B Q \right)^\top p \right\|_*. \quad (2.3.3)$$

For simplicity, we follow [11] by approximating the integral in (2.2.12) with a left Riemann sum quadrature with N equally spaced terms defined by

$$t_i = i\Delta t, \quad (2.3.4)$$

with $i \in \{0, \dots, N-1\}$ and $\Delta t = \frac{T}{N}$. The generalized Hopf formula, for some terminal time T , becomes

$$\begin{aligned} \varphi(z_0, T) = & -\min_{p \in \mathbb{R}^n} \left\{ J_z^*(p, 0) \right. \\ & + \Delta t \sum_{i=0}^{N-1} \left\| (-e^{-(T-t_i)A} BQ)^\top p \right\|_* \\ & \left. - \langle z_0, p \rangle \right\}. \end{aligned} \quad (2.3.5)$$

Other forms of quadrature can be used for integral approximations, such as trapezoidal or Simpson's rule [24], that increase approximation accuracy without sacrificing computational performance. Additionally, other advanced approximations could be considered, such as those employed by pseudospectral methods in [25], and will be investigated in future work.

To formulate as a primal-dual optimization, first set

$$G(p) = J_z^*(p, 0) - \langle z_0, p \rangle. \quad (2.3.6)$$

Now define

$$K_i = (-e^{-(T-t_i)A} BQ)^\top \quad (2.3.7)$$

and $F_i = \Delta t \|(\cdot)\|_* = (\mathcal{I}_{\Delta t u})^*$, where we denote by $\mathcal{I}_{\alpha \mathcal{C}}$ the indicator function of a set \mathcal{C} scaled by a constant $\alpha > 0$ defined as

$$\mathcal{I}_{\alpha \mathcal{C}}(x) = \begin{cases} 0 & \text{if } \frac{x}{\alpha} \in \mathcal{C} \\ +\infty & \text{otherwise.} \end{cases}$$

The sum term in (2.3.5) can be written as

$$\Delta t \sum_{i=0}^{N-1} \left\| \left(-e^{-(T-t_i)A} B Q \right)^\top p \right\|_* = \sum_{i=0}^{N-1} F_i(K_i p).$$

We can now form the matrix K as

$$K = \begin{bmatrix} K_1 \\ K_2 \\ \vdots \\ K_N \end{bmatrix},$$

which gives

$$F(Kp) = \sum_{i=0}^{N-1} F_i(K_i p), \quad (2.3.8)$$

and combined with (2.3.6) is now in the form of (2.3.1). Note that K in this formulation is non square (preconditioning [26] can be used to enhance convergence rate). If the action of the matrix exponential in (2.3.7) is not known, then it can be quickly evaluated, for all time samples, without resorting to computing the matrix exponential with [27]. The structure of (2.3.8) forms a *separable sum*. This implies that for some $y = (y_1, y_2, \dots, y_N)$

$$F^*(y_1, y_2, \dots, y_N) = F_1^*(y_1) + F_2^*(y_2) + \dots + F_N^*(y_N).$$

Recall that $F_i = (\mathcal{I}_{\Delta t} u)^*$, thus

$$F_i^*(y_i) = \mathcal{I}_{\Delta t} u. \quad (2.3.9)$$

The proximal operator of a separable sum is simply

$$(I + \partial F^*)^{-1}(y_1, y_2, \dots, y_N) = \begin{bmatrix} (I + \partial F_1^*)^{-1}(y_1) \\ (I + \partial F_2^*)^{-1}(y_2) \\ \vdots \\ (I + \partial F_N^*)^{-1}(y_N) \end{bmatrix}. \quad (2.3.10)$$

The proximal operators in (2.3.10) are independent of each other, and as a result can be computed in parallel. This can be advantageous for real-time implementations in hardware such as multi-core embedded CPUs and field programmable gate arrays (FPGAs).

2.3.2 Stopping Criteria

Care must be taken to select the stopping criterion of the algorithm in (2.3.2). The step sizes τ and σ are in general not equal and any stopping criteria must account for this. As a result we choose the primal and dual residuals [28, See Eqs. 10 and 12] as a step-size dependent stopping criteria and stop iterating (2.3.2) when the conditions

$$\|(p^k - p^{k-1})/\tau - K^\top (y^k - y^{k-1})\| < \varepsilon,$$

and

$$\|(y^k - y^{k-1})/\sigma - K (p^k - p^{k-1})\| < \varepsilon,$$

are both met. For all the experiments listed in Section 2.6, ε is set to 10^{-4} .

2.4 Time-Optimal Control

To find the time-optimal control to some convex terminal set Ω , choose a convex terminal cost function J_x such that

$$\begin{cases} J_x(x) < 0 & \text{for any } x \in \text{int } \Omega, \\ J_x(x) > 0 & \text{for any } x \in (\mathbb{R}^n \setminus \Omega), \\ J_x(x) = 0 & \text{for any } x \in (\Omega \setminus \text{int } \Omega), \end{cases}$$

where $\text{int } \Omega$ denotes the interior of Ω . The intuition behind defining the terminal cost function this way is simple. If the value function $\varphi(x_0, T) < 0$ for some x_0 and T , then there exists a control $u(t)$ that drives the state from the initial condition at x_0 , to the final state, $x(T)$ inside the set Ω . The smallest value of time T , such that $\varphi(x_0, T) = 0$ is the minimum time to reach the set Ω , starting at state x_0 . Recall that the initial value function, $J_z(p, 0)$, and its associated Fenchel–Legendre transform, $J_z^*(p, 0)$ of the Hopf formula in (2.2.12), is defined in z , and must be transformed with (2.2.11). The minimum time to reach Ω is denoted by T^* , and the control computed at T^* is the time-optimal control. As first noted in [29], Hopf formula is itself a Fenchel–Legendre transform. It follows from a well known property of the Fenchel–Legendre transform [30] that the unique minimizer of (2.2.12) is the gradient of the value function

$$\begin{aligned} \nabla_z \varphi(z_0, T) = \arg \min_{p \in \mathbb{R}^n} & \left\{ J_z^*(p, 0) \right. \\ & \left. + \int_0^T H_z(p, s) ds - \langle z_0, p \rangle \right\}, \end{aligned} \tag{2.4.1}$$

provided the gradient exists. So when solving for the value function using (2.2.12), we automatically solve for the gradient. We will refer to the minimizer in (2.4.1) as

$$p^* = \nabla_z(z_0, T).$$

We propose solving for the minimum time to reach the set Ω, T^* , by a hybrid method of the bisection method and Newton's method. Newton's method has been shown to have faster convergence (quadratic) than bisection, but is unstable when the gradient is small and motivates the use of a hybrid method. We can iterate time, t_{n+1} , with Newton's as

$$t_{n+1} = t_n - \frac{\varphi(x_0, t_n)}{\frac{\partial \varphi}{\partial t}(x_0, t_n)}. \quad (2.4.2)$$

As noted in [10], $\frac{\partial \varphi}{\partial t}(x_0, t)$ must satisfy the Hamilton–Jacobi equation (2.2.4). Therefore we have

$$\frac{\partial \varphi}{\partial t}(x_0, t_n) = -H_x(\nabla_x \varphi(x_0, t_n), x_0).$$

We also see from (2.2.9) and applying the chain rule that

$$\begin{aligned} \nabla_x \varphi(z(t), T-t) &= \nabla_x \varphi(e^{-tA}x(t), T-t) \\ &= e^{-tA^\top} \nabla_x \varphi(z(t), T-t). \end{aligned}$$

Therefore when $t = 0$, then $z_0 = x_0$, $\varphi(z_0, T) = \varphi(x_0, T)$, and

$$\nabla_x(x_0, T) = \nabla_z(z_0, T).$$

This implies that (2.4.2) can be written as

$$t_{n+1} = t_n + \frac{\varphi(z_0, t_n)}{H_x(\nabla_z \varphi(z_0, t_n), z_0)}. \quad (2.4.3)$$

For the purpose of evaluating (2.4.3), there is no need to apply the change of variables

as in (2.2.9). Therefore we have

$$H_x(p^*, z_0) = -z_0^\top A^\top p^* + \|-Q^\top B^\top p^*\|_*.$$

If it is known that a single zero exists on the interval $T^* \in [0, t_{max}]$, then we use the Newton update from (2.4.3) to find t_{n+1} . With the value function computed at each Newton iteration, we can keep track of the updated interval $T^* \in [t_{min}, t_{max}]$ and use a bisection update if the Newton update of t_{n+1} is out side this interval. Once the minimum time to reach the set Ω , T^* , is found, the optimal control $u^*(t)$ can be found from the relation

$$\begin{aligned} \nabla_p H_z(\nabla_z \varphi(z_0, T^*), T^*) &= \nabla_p H_z(p^*, T^*) \\ &= e^{-(T^*-t)A} B u^*(t). \end{aligned} \tag{2.4.4}$$

2.5 Trajectory Generation with the Generalized Hopf Formula

Using the solution of the Hopf formula, we illustrate a dynamic programming point of view [9, 18] of the associated Hamilton–Jacobi equation to compute the optimal trajectory. We denote by $\gamma_z(s) \in \mathbb{R}^n$, with $s \in (0, T)$, as the state trajectory with $\gamma_z(0) = z_0$. Recall the fact that the solution of short-time Hopf formula (2.2.6) is itself a Fenchel transform [29]

$$\varphi(z, t) = (J^* + tH)^*(z). \tag{2.5.1}$$

Note that the minimizer of $J_z^*(p)$ is the optimal terminal state $\gamma_z(T)$, and, if J_z^* is differentiable, then can be found with

$$\gamma_z(T) = \frac{\partial}{\partial p} J_z^*(p^*, 0), \quad (2.5.2)$$

where p^* is the solution to the Hopf formula in (2.2.6). Now consider the case where p^* is the solution to the generalized Hopf formula given in (2.2.12), and what follows is an analysis in the variable z . This implies that

$$\varphi(z_0, T) = - \left\{ J_z^*(p^*, 0) + \Delta t \sum_{i=0}^{N-1} H_z(p^*, t_i) - \langle z_0, p^* \rangle \right\}, \quad (2.5.3)$$

where each of the quadrature time samples t_i are equally spaced by Δt seconds on the interval $[0, T]$ as defined by (2.3.4). Recall from Section 2.2.2 that we denote by $\varphi_z^*(p, T)$ the Fenchel transform of $\varphi(z, t)$ with respect to the variable z at time $t = T$. If we write the Hopf formula with initial convex data $J(z, T - \Delta t) = \varphi(z, T - \Delta t)$, then we can find the level set evolution only for a short time, Δt , starting at value $\varphi(z, T - \Delta t)$ with

$$\begin{aligned} \varphi(z_0, T) &= - \left\{ \varphi_z^*(p^*, T - \Delta t) \right. \\ &\quad \left. = +\Delta t H_z(p^*, T - \Delta t) - \langle z_0, p^* \rangle \right\}. \end{aligned}$$

Following (2.5.2), the optimal state, with respect to z is

$$\gamma_z(\Delta t) = \frac{\partial}{\partial p} \varphi_z^*(p^*, T - \Delta t). \quad (2.5.4)$$

From (2.5.1) we conclude that

$$\begin{aligned}
 \varphi_z^*(p^*, T - \Delta t) &= (J^* + \Delta t H_z)_p^{**}(z, T - \Delta t) \\
 &= J_z^*(p^*, T - 2\Delta t) \\
 &\quad + \Delta t H_z(p^*, T - 2\Delta t).
 \end{aligned} \tag{2.5.5}$$

The last line in (2.5.5) is due to the fact that if f is convex, proper and lower semicontinuous, then $f^{**} = f$. We form a recursive operation, repeating (2.5.5) until we reach time zero and get

$$\begin{aligned}
 \varphi(z_0, T) &= - \left\{ \varphi_z^*(p^*, 0) \right. \\
 &\quad \left. + \Delta t \sum_{i=0}^{N-1} H_z(p^*, t_i) - \langle z_0, p^* \rangle \right\},
 \end{aligned}$$

which is equivalent to (2.5.3) since by definition $J(x, 0) = \varphi(x, 0)$. This suggests that the generalized Hopf formula with the integral in (2.2.12) approximated by quadrature is equivalent to the composition of many short-time Hopf formulas (2.2.6) of length Δt . Also, the recursion can be applied to the optimal terminal point from (2.5.4) as

$$\begin{aligned}
 \gamma_z(\Delta t) &= \frac{\partial}{\partial p} \varphi_z^*(p^*, T - \Delta t) \\
 &= \frac{\partial}{\partial p} \left\{ J_z^*(p^*, 0) + \Delta t \sum_{i=0}^{N-1} H_z(p^*, t_i) \right\}.
 \end{aligned}$$

This can equivalently be used for the optimal trajectory at any time sample t_i as

$$\begin{aligned} \gamma_z(t_i) &= \frac{\partial}{\partial p} \left\{ J_z^*(p^*, 0) + \Delta t \sum_{k=0}^i H_z(p^*, t_k) \right\} \\ &= \frac{\partial}{\partial p} J_z^*(p^*, 0) + \Delta t \sum_{k=0}^i \frac{\partial}{\partial p} H_z(p^*, t_k). \end{aligned} \quad (2.5.6)$$

If we are interested in the trajectory at each quadrature time sample, we don't have to recompute the sum for each t_k , since we can incrementally build the trajectory point from t_{k+1} . Note that each point in (2.5.6) is in terms of the state variable z , and can be found for x by applying inverse of the transform given in (2.2.9). Non-rigorously we see that the time rate of change of the state trajectory in (2.5.6) is equal to the gradient with respect to p of the Hamiltonian, which satisfies Pontryagin's Maximum Principle [31], though it was derived using only the Hopf formula and basic principles of convex analysis.

2.6 Results

The primal-dual method presented in Section 2.3 was implemented in MATLAB R2017a on a laptop equipped with an Intel Core i7-7500 CPU running at 2.70 GHz. For all experiments, 100 time samples were used for the quadrature in (2.3.5) and the dual step size was set to $\sigma = \frac{1}{\tau \|K\|^2}$, and $\theta = 1$. The initial conditions for all examples is $p^0 = \bar{p}^0 = x_0$ for the primal variable and $y^0 = Kp^0$ for the dual variable. Note that for some of the examples to follow, the action of the matrix exponential is known in closed form and that can be used for increased computational enhancement. Since in general this is not the case, we used [27] to numerically compute the action of the matrix exponential as to show dimensional scaling properties even for the most general case.

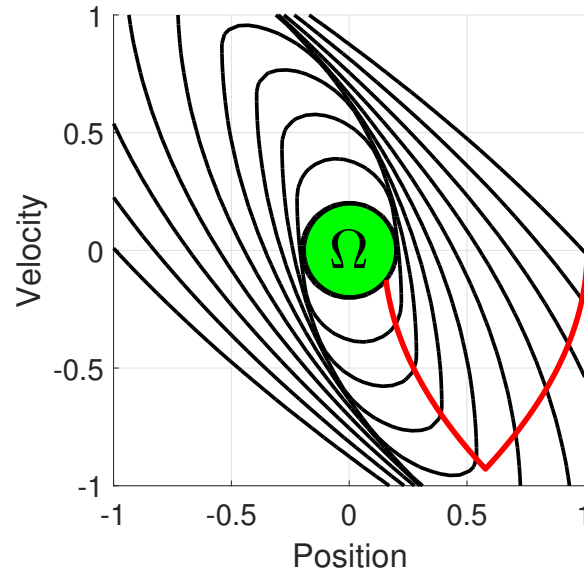


Figure 2.6.1: The double integrator example. The zero level set evolution solved at various times shown in black. Direct trajectory generation shown in red. The terminal set, Ω , is shaded green.

2.6.1 Double Integrator

We begin with the simple double integrator problem with system $\dot{x} = Ax + Bu$ with

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

where the state $x \in \mathbb{R}^2$ is position and velocity. This problem is selected since closed form optimal solutions exist and is low enough dimension to compare the level set evolution to that of grid based numeric techniques such as that in [6, 32]. We consider the control to be constrained to $u \in [-1, 1]$ and implies a control of $\mathcal{C} = \{u : |u| \leq 1\}$. After a change

of variables (2.2.9), the Hamiltonian becomes

$$H_z(p, t) = \left| \left(-e^{-(T-t)A} B \right)^\top p \right|.$$

We chose the terminal set to be an ellipsoidal with

$$\Omega = \{x : \langle x, W^{-1}x \rangle \leq 1\}. \quad (2.6.1)$$

where W is symmetric positive definite. For the initial cost function J_x , the elements of W are selected such that Ω is a circle with radius $r = 0.2$. The terminal cost function becomes

$$J_z(z, 0) = \langle z, V(0)z \rangle - 1$$

where $V(t) = e^{(T-t)A^\top} W^{-1} e^{(T-t)A}$. This gives

$$J_z^*(p, 0) = \frac{1}{4} \langle p, V(0)^{-1} p \rangle + 1. \quad (2.6.2)$$

In this example (2.3.6) becomes

$$G(p) = \frac{1}{4} \langle p, V(0)^{-1} p \rangle - \langle z_0, p \rangle + 1, \quad (2.6.3)$$

which is quadratic and results in the following proximal point of G at p :

$$(I + \tau \partial G)^{-1}(p) = \left(I + \frac{1}{2} \tau V(0)^{-1} \right)^{-1} (p + \tau z_0). \quad (2.6.4)$$

Note that we do not need to compute the inverse of $V(0)$ since

$$V(0)^{-1} = e^{-TA} W e^{-TA^\top}.$$

Likewise, with F_i^* defined in (2.3.9), the proximal points of each F_i^* at y_i in (2.3.10) is given by

$$(I + \sigma \partial F_i^*)^{-1}(y_i) = \Delta t \operatorname{sign}(w_i) \min(|w_i|, 1),$$

where $w_i = \frac{y_i}{\Delta t}$ and $\operatorname{sign}(\beta) = 1$ if $\beta \geq 0$ and -1 otherwise.

We computed the solution to the Hamilton–Jacobi equation at each point on a grid, $[-1, 1]^2$, of 50 equally spaced points in each dimension. This was done to average execution time for a large number of initial conditions. The average computational time was 2.1 *ms* per point on the grid and the zero contours of the value function for ten different times equally spaced on $t \in [0, T^*]$ are shown in Figure 2.6.1. The value of T^* we set as by solving for the minimum time to the zero level set for the initial state $z_0 = x_0 = (1, 0)^\top$ using the method presented in Section 2.4. The primal step size was set to $\tau = 10$.

The optimal trajectory starting at z_0 was computed following (2.5.6) with

$$\gamma_z(T^*) = \frac{\partial}{\partial p} J_z^*(p^*, 0) = \frac{1}{2} V(0)^{-1} p^*,$$

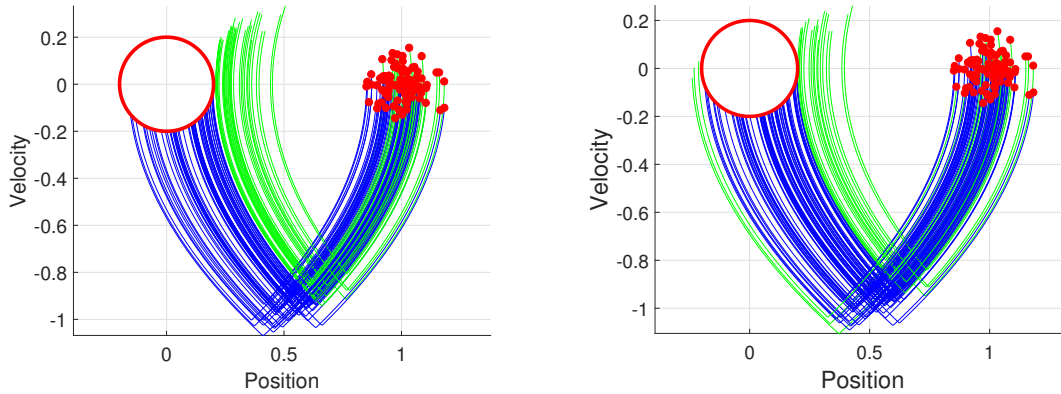
since $J_x(x)$ is quadratic. The gradient of the Hamiltonian is

$$\begin{aligned} \frac{\partial}{\partial p} H_z(p^*, t_k) &= -e^{-(T^*-t_k)A} B \\ &\times \operatorname{sign}\left(\left(-e^{-(T^*-t_k)A} B\right)^\top p^*\right). \end{aligned}$$

The trajectory as computed in (2.5.6) is shown in red in Figure 2.6.1.

2.6.2 Unscented Optimal Control

We can utilize the favorable dimensional scaling of the proposed methods to generate solutions that are robust to system uncertainty. These uncertainties could be the initial



(a) 100 random samples of initial conditions with the existing control formulation. 66 samples reached the goal state and 34 missed.

(b) 100 random samples of initial conditions with uncertainty modeled as an unscented control problem. 79 samples reached the goal state and 21 missed.

Figure 2.6.2: An example of unscented control applied to a time-optimal double integrator problem with uncertain initial state. Red are samples of the initial states. Blue trajectories are random initial states that reached the goal state and green are trajectories that missed the goal state. Best viewed in color.

state or system parameters such as mass, lift coefficients, or other properties. The idea is to augment the system to include samples that represent different initial conditions or parameters that share one, common control input. This system forms a tyochastic [33] differential equation, since the parameters are fixed, but unknown at run time. The goal is to select a single control so the aggregate cost of all samples is optimized.

For uncertainties that can be modeled as Gaussian, we can choose the samples deterministically with the unscented transform [34], and these samples are typically referred to in literature as sigma points. This transform provides a second-order approximation of the moments of a Gaussian distribution propagated through a nonlinear function. It was developed for state estimation problems and in this context is known as the unscented Kalman filter [35, 36]. Using the unscented transform for sample selection to approximate the tyochastic optimal control problem was developed by Ross et al. [37, 38]. The extra state dimensions that result from this technique is not as problematic with

proximal splitting as it may be with other methods.

Let $x = (\chi^{[1]}, \chi^{[2]}, \dots, \chi^{[2n+1]})^\top$, with $\chi^{[i]} \in \mathbb{R}^n$ represent the new state vector augmented by $2n + 1$ states generated from the unscented transform and the new system becomes

$$\dot{x} = \begin{bmatrix} \dot{\chi}_1 \\ \dot{\chi}_2 \\ \vdots \\ \dot{\chi}_{2n+1} \end{bmatrix} = \begin{bmatrix} f_1(\chi^{[1]}, u) \\ f_2(\chi^{[2]}, u) \\ \vdots \\ f_{2n+1}(\chi^{[2n+1]}, u) \end{bmatrix},$$

subject to initial condition $z_0 = x_0 = (\chi_0^{[1]}, \chi_0^{[2]}, \dots, \chi_0^{[2n+1]})$. Take for example the problem of uncertainty in initial condition, $x_0 \sim \mathcal{N}(\mu, \Sigma)$. As an unscented control problem, the dynamics are the same, so $f_1 = f_2 = \dots = f_{2n+1} = f$, and our augmented system is $\dot{x} = \tilde{A}x + \tilde{B}u$ with

$$\tilde{A} = \begin{bmatrix} A & \cdots & 0 \\ \vdots & A & \vdots \\ & & \ddots \\ 0 & \cdots & A \end{bmatrix},$$

and

$$\tilde{B} = \begin{bmatrix} B \\ B \\ \vdots \\ B \end{bmatrix}.$$

The initial state becomes $x_0 = \left\{ \chi_0^{[i]} \right\}_{i=1, \dots, 2n+1}$, where each $\chi_0^{[i]}$ is formed by the unscented transform with mean μ and covariance Σ . The mean square error of the terminal state

relative to some goal state, \hat{x}_T is found with the unscented transform by

$$\text{MSE} \approx \sum_{i=1}^{2n+1} w_m^{[i]} \left(\chi_T^{[i]} - \hat{x}_T \right)^\top \left(\chi_T^{[i]} - \hat{x}_T \right), \quad (2.6.5)$$

where w_m is the mean weight factor¹, and $\chi_T^{[i]}$ is the terminal state for the i -th unscented sigma point. For this example, we wish to find the minimum time to reach the origin subject to the constraint that the mean square error is less than some threshold ℓ . This can be found from (2.6.5) with

$$\text{MSE} = \mathbb{E} \left[(\hat{x}_T - x(T))^\top (\hat{x}_T - x(T)) \right] \leq \ell.$$

We select the origin as the goal with $\hat{x} = (0, 0)^\top$, and formulate as an unscented control problem. The trace of the terminal covariance can be represented by the quadratic

$$J(z) = \langle z, V(0)z \rangle - \ell \quad (2.6.6)$$

with $V(t) = e^{(T-t)A^\top} W^{-1} e^{(T-t)A}$ and

$$W = \begin{bmatrix} \left(w_m^{[1]} \right)^{-1} I_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \left(w_m^{[2n+1]} \right)^{-1} I_2 \end{bmatrix},$$

where I_2 is the 2×2 identity matrix.

Figure 2.6.2 shows example trajectories when the initial condition has random perturbations, with $\mu = x_0$ and $\Sigma = \pi^2 I_2$, for a double integrator problem. The standard deviation was set to $\pi = 0.0667$. If the initial state is *exactly* what was used to compute

¹For more information on the generation of the unscented sigma points, and their weights, see [36].

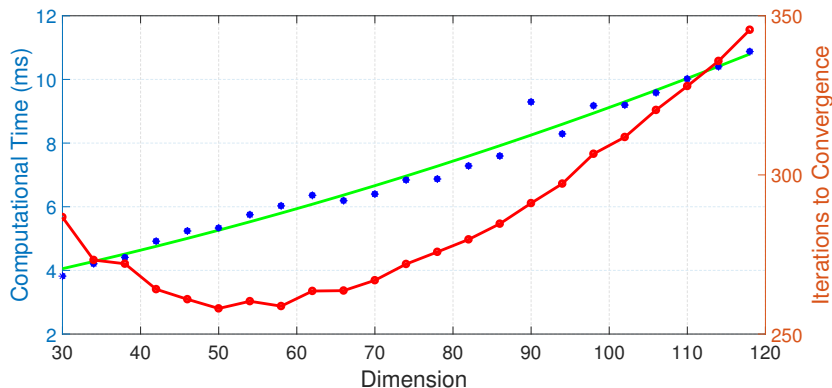


Figure 2.6.3: Dimensional scaling properties as described by the experiment in Section 2.6.3. The blue dots represent average computational time for an experiment and the green line is the least squares polynomial fit to computational time. The red displays average iterations to convergence.

the optimal control, then the trajectory generated reaches the goal and is time-optimal. However, if the initial state is perturbed, some trajectories miss the intended goal entirely. In this particular example, it is especially sensitive to perturbations to the “right” in the spatial x direction and is shown in the Figure 2.6.2a on the left. Of 100 random initial states, 34 miss the goal state. For the resulting 10 dimensional unscented control problem with $\tau = 0.5$, the HJ solutions were found on average 4.0 ms per point. Trajectory samples are shown in the Figure 2.6.2a on the right, the number of trajectories that miss the goal is reduced to only 5.

2.6.3 Dimensional Scaling

Next we seek to analyze how the proposed method scales with dimension. We can construct a problem similar to that presented in Section 2.6.2 but by selecting samples at random as opposed to using the unscented transform. Constructing a problem in this fashion is something not typically done in practice, but allows us to vary the number of random samples, and hence alter the dimension of the problem in a consistent and uniform way. The initial state for k samples becomes $x_0 = \left\{ \chi_0^{[i]} \right\}_{i=1, \dots, k}$, where each $\chi_0^{[i]}$

is an independent and identically distributed (iid) random vector drawn according to $\chi_0^{[i]} \sim \mathcal{N}(\mu, \Sigma)$. To again penalize the trace, the terminal cost function is the same as (2.6.6) except W is now defined by

$$W = \begin{bmatrix} kI_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & kI_2 \end{bmatrix}.$$

Figure 2.6.3 shows the average computational time and average iterations to convergence for dimensions ranging from 30 to 120. The green line in the figure is the least squares polynomial fit for average computation time in milliseconds. If we let d denote problem dimension for the experiments, the fit is $t_{\text{comp}} = 2.382 \times 10^{-4}d^2 + 0.0414d + 2.598$. Note the extremely small quadratic coefficient.

2.7 Conclusion

Presented is a parallel primal-dual method to solve the Hamilton–Jacobi equation for time-optimal control using the generalized Hopf formula. We empirically showed how the method scales approximately quadratic with dimension, though with small quadratic coefficient. The experiments were shown using Matlab, and simple implementation in a compiled language could provide significant computational improvement. Future work includes increased experimentation with different systems, the use of more advanced quadrature methods, adaptive step sizes, time-varying systems, and state dependent Hamiltonians.

Chapter 3

Time-Optimal Collaborative Guidance Using the Generalized Hopf Formula

3.1 Introduction

One of the first successful implementations of control laws for pursuit problems is proportional navigation (PN) [39], which attempts to drive the rate of the line-of-sight vector between pursuer and evading target vehicle to zero. In this derivation, the target vehicle is assumed moving, but not maneuvering (turning). Generalizations of this concept attempt to estimate the vehicle maneuver [12], but these methods are not optimal since evasion strategy is not considered, i.e. not formulated as a differential game [40]. Additionally, this family of control laws does not account for control saturation. PN typically requires the magnitude of the control bound of the pursuer to be much greater than that of the evader to be successful, on the order of 3-5 times greater [12]. These guidance laws are strictly one-on-one in nature, and do not readily generalize to

collaborative systems of multiple vehicles where the desired pursuit guidance is to 'team' together to capture a target. These early pursuit problems typically referred to controller designs as *guidance laws*, and in this letter we will use the terms controller and guidance interchangeably.

More recently, [41] proposed a solution to multi-vehicle pursuit evasion in a plane. In this case the problem was solved sub-optimally with heuristics in an effort to avoid the computational burden of direct solution to the Hamilton-Jacobi equation. Additionally, the method was based on simplified, single-integrator dynamics that require the vehicles to maneuver instantaneously to ensure capture.

A general alternative is to formulate the pursuit-evasion problem as a differential game, and derive a Hamilton–Jacobi–Isaacs (HJI) equation representing the optimal cost-to-go of the system. Traditionally, numerical solutions to HJI equations require a dense, discrete grid of the solution space [5, 6, 7]. Computing the elements of this grid scales poorly with dimension and has limited use for problems with dimension of greater than four. The exponential dimensional scaling in optimization is sometimes referred to as the “curse of dimensionality” [8, 9]. This phenomenon is seen clearly in [42], which formulated a differential game for a capture-the-flag problem and solved numerically on a four dimensional grid with [32]. The computational time was as much as 4 minutes, too slow for real-time application, even with a coarsely sampled grid of 30 points in each dimension and with low numeric accuracy. When the grid is increased to 45 points in each dimension and with high numeric accuracy, the computation time jumps to an hour.

Recent research [10] has discovered numerical solutions based on the generalized Hopf formula that do not require a grid and can be used to efficiently compute solutions to a certain class of Hamilton–Jacobi equations that arise in linear control theory and differential games. This readily allows the generalization with pursuit-evasion to collaborative guidance of multiple pursuing vehicles.

This letter presents a new method for multi-vehicle collaborative pursuit guidance of a maneuvering target, showing that teams of vehicles can intercept the target without requiring drastically higher control bound as in the family of methods in [12]. A joint system state space representing the kinematics of all pursuing vehicles relative to the target was constructed, the dimension of which makes it infeasible for traditional grid-based methods. This high-dimensional problem was then efficiently solved using the generalized Hopf formula, and included the constraint of time-varying bounds on the magnitude of available vehicle control, while ensuring intercept when starting within the reachable set.

The rest of the paper is organized as follows. We derive the models used in the study in Sec. 3.2 followed the presentation of efficient solution techniques that employ the generalized Hopf formula to solve the Hamilton–Jacobi equations for optimal control and differential games in Sec. 3.3. The application of these methods to collaborative guidance is given in Sec. 3.4, followed by results on a planar, multiple vehicle pursuit-evasion game in Sec. 3.5.

3.2 Pursuit-Evasion Model

3.2.1 Single Vehicle Model

First consider the pursuit-evasion game with only a single pursuer. We construct a state space representation of the position and orientation of the pursuer relative to the evader, with geometry shown in Fig. 3.2.1. With $x = [\delta x, \delta y, \delta \theta]^\dagger$, the relative system

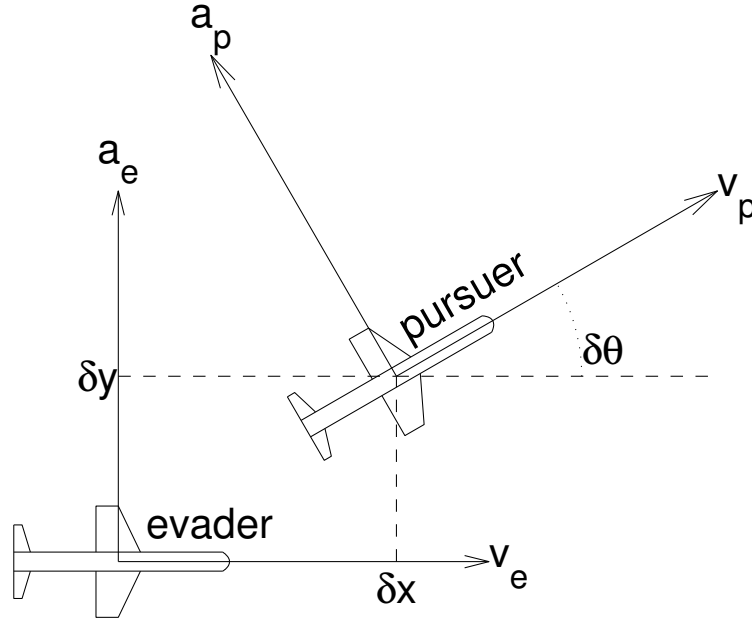


Figure 3.2.1: The engagement geometry of the system presented in (3.2.1).

becomes

$$\dot{x}(t) = \begin{bmatrix} V_p \cos(\delta\theta) - V_e + \frac{\delta y a_e}{V_e} \\ V_p \sin(\delta\theta) - \frac{\delta x a_e}{V_e} \\ \frac{a_p}{V_p} - \frac{a_e}{V_e} \end{bmatrix}, \quad (3.2.1)$$

with V_p and V_e representing the forward speed of the pursuer and evader, respectively. The terms a_p and a_e are the lateral acceleration inputs to the system for each vehicle. These accelerations are limited by the physical maneuver capabilities of the vehicles. This system is based on the relative state [7] of two modified Dubin's car [43, 44] models, with acceleration instead of the more common turn rate input. Additionally, we constructed this system to be evader centric, allowing for the addition of multiple pursuers. Denoting by \dagger the transpose of a matrix, we introduce the new state vector $x = [\delta x, \delta y, \delta v_x, \delta v_y]^\dagger$, where δx and δy are the positional displacement separating the vehicles (see Figure 3.2.1), $\delta v_x = V_p - V_e$, and δv_y is the relative vertical velocity. We proceed to linearize the system

(3.2.1) with

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} 0_2 & I_2 \\ 0_2 & 0_2 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \pm 1 \end{bmatrix} a_p + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} a_e, \\ &= Ax(t) + Ba_p + Da_e, \end{aligned} \quad (3.2.2)$$

with the \pm sign needed depending on whether its tail-chase (+) or head-on (−) engagement. The linearization at first glance may seem extreme, but this linearization strategy is used when deriving proportional navigation, or its variants such as augmented proportional guidance and extended proportional guidance, using linear quadratic control techniques [12]. The controls for the pursuer are constrained to the set $\mathcal{A}_p = \{a_p : \|Q_p^{-1}(t)a_p\|_\infty \leq 1\}$ and the controls for the evader are constrained to the set $\mathcal{A}_e = \{a_e : \|Q_e^{-1}a_e\|_\infty \leq 1\}$. The infinity norm with diagonal matrix Q , scales the control limit independently in orthogonal directions. Q_p is a function of time since some systems have control bounds that vary with time, and is needed to model aerodynamic control surfaces on decelerating vehicles. Both controls are considered symmetric (centered at zero) for this paper and all simulations.

We represent the capture set, Ω , as an ellipsoid

$$\Omega = \{x : \langle x, W^{-1}x \rangle \leq 1\}. \quad (3.2.3)$$

where W is the ellipsoid shape matrix. The elements of W are selected such that the

pursuing vehicle must be within a distance r

$$\left\| \begin{bmatrix} \delta x \\ \delta y \end{bmatrix} \right\| \leq r,$$

and the velocity at intercept is within some large bound V_{\max} (we don't care what the velocity was at capture, just as long as capture has occurred). This gives

$$W = \begin{bmatrix} r^2 & \cdots & 0 \\ & r^2 & \vdots \\ \vdots & & V_{\max}^2 \\ 0 & \cdots & & V_{\max}^2 \end{bmatrix}.$$

3.2.2 Multi-Vehicle Model

For a multi-vehicle problem with k pursuers against a single evader, the joint state space with state vector $\chi \in \mathbb{R}^{4 \times k}$ can be constructed as follows

$$\chi = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_k \end{bmatrix} = \begin{bmatrix} A & \cdots & 0 \\ & A & \vdots \\ \vdots & & \ddots \\ 0 & \cdots & A \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} + \begin{bmatrix} B_1 & \cdots & 0 \\ & B_2 & \vdots \\ \vdots & & \ddots \\ 0 & \cdots & B_k \end{bmatrix} \begin{bmatrix} a_{p1} \\ a_{p2} \\ \vdots \\ a_{pk} \end{bmatrix} + \begin{bmatrix} D \\ D \\ \vdots \\ D \end{bmatrix} a_e \quad (3.2.4)$$

$$\implies \dot{\chi} = \hat{A}\chi + \hat{B}a_p + \hat{D}a_e. \quad (3.2.5)$$

Collaborative guidance is induced by noticing that capture can happen by *any* single vehicle of the k vehicles in the system. The capture set for the i -th vehicle in the joint system (3.2.4) is denoted as

$$\Omega_i = \{ \chi : \langle \chi, W_i^{-1} \chi \rangle \leq 1 \},$$

with the shape matrix defined as the block diagonal matrix with W on the i -th block of the matrix, and the 4×4 matrix $\Sigma = V_{\max}^2 I$ occupying all other blocks. This implies

that the capture set for the joint system is

$$\Omega = \cup_i \Omega_i. \quad (3.2.6)$$

3.3 Hamilton–Jacobi Equations with Bounded Control

3.3.1 Viscosity Solutions with the Hopf Formula

To compute optimal guidance, we use the generalized Hopf formula [10, 19, 29]. Consider system dynamics represented as

$$\dot{x}(t) = f(u(t)) \quad (3.3.1)$$

where $x(t) \in \mathbb{R}^n$ is the system state and $u(t) \in \mathcal{C} \subset \mathbb{R}^m$ is the control input, constrained to lie in the convex admissible control set \mathcal{C} . We consider a cost functional for a given initial time t , and terminal time T

$$K(x, t, u) = \int_t^T L(u(s)) ds + J(x(T)), \quad (3.3.2)$$

where $x(T)$ is the solution of (3.3.1) at terminal time, T . We assume that the terminal cost function $J : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex. The function $L : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is the running cost, and is assumed proper, lower semicontinuous, convex, and 1-coercive. The value function $v : \mathbb{R}^n \times (-\infty, T] \rightarrow \mathbb{R}$ is defined as the minimum cost, K , among all admissible

controls for a given state x , and time $t \leq T$ with

$$v(x, t) = \inf_{u \in \mathcal{C}} K(x, t, u). \quad (3.3.3)$$

The value function in (3.3.3) satisfies the dynamic programming principle [17, 18] and also satisfies the following initial value Hamilton-Jacobi (HJ) equation by defining the function $\varphi : \mathbb{R}^n \times \rightarrow \mathbb{R}$ as $\varphi(x, t) = v(x, T - t)$, with φ being the viscosity solution of

$$\begin{cases} \frac{\partial \varphi}{\partial t}(x, t) + H(t, \nabla_x \varphi(x, t)) = 0 & \text{in } \mathbb{R}^n \times (0, +\infty), \\ \varphi(x, 0) = J(x) & \forall x \in \mathbb{R}^n, \end{cases} \quad (3.3.4)$$

where the Hamiltonian $H : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is defined by

$$H(p) = \sup_{c \in \mathbb{R}^m} \{\langle -f(c), p \rangle - L(c)\}. \quad (3.3.5)$$

To apply the constraint that the control must be bounded, we introduce the following running cost $L = \mathcal{I}_{\mathcal{C}}$, where

$$\mathcal{I}_{\mathcal{C}} = \begin{cases} 0 & \text{if } c \in \mathcal{C} \\ +\infty & \text{otherwise,} \end{cases}$$

is the indicator function for the set \mathcal{C} . This induces a time-optimal control formulation and reduces the Hamiltonian to

$$H(p) = \max_{c \in \mathcal{C}} \langle -f(c), p \rangle.$$

Solving the HJ equation (3.3.4) describes how the value function evolves with time at any point in the state space and from this, optimal control policies can be found.

It was shown in [10] that an exact, point-wise viscosity solution to (3.3.4) can be found

using the Hopf formula [19]. The value function can be found with the Hopf formula

$$\varphi(x, t) = -\min_{p \in \mathbb{R}^n} \{J^*(p) + tH(p) - \langle x, p \rangle\}, \quad (3.3.6)$$

where the Fenchel-Legendre transform $g^* : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ of a convex, proper, lower semicontinuous function $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is defined by [45]

$$g^*(p) = \sup_{x \in \mathbb{R}^n} \{\langle p, x \rangle - g(x)\}. \quad (3.3.7)$$

Following the basic definition of the Fenchel-Legendre transform, (3.3.6) can be written [29] as

$$\varphi(x, t) = (J^* + tH)^*(x).$$

This shows that value function is itself a Fenchel-Legendre transform. It follows from a well known property of the Fenchel-Legendre transform [30] that the unique minimizer of (3.3.6) is the gradient of the value function

$$\nabla_x \varphi(x, t) = \arg \min_{p \in \mathbb{R}^n} \{J^*(p) + tH(p) - \langle x, p \rangle\},$$

provided the gradient exists. So by solving for the value function using (3.3.6), we automatically solve for the gradient.

3.3.2 General Linear Models

Now consider the following linear state space model

$$\dot{x}(t) = Ax(t) + B(t)u(t), \quad (3.3.8)$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, state vector $x \in \mathbb{R}^n$, and control input $u \in \mathcal{C} \subset \mathbb{R}^m$. We can make a change of variables

$$z(t) = e^{-tA}x(t), \quad (3.3.9)$$

which results in the following system

$$\dot{z}(t) = e^{-tA}B(t)u(t), \quad (3.3.10)$$

with terminal cost function now defined in z with $\varphi(z, 0) = J_z(z, 0) = J_x(e^{TA}z)$, which depends on terminal time, T . Notice that the system is of the form presented in (3.3.1), with the exception that the system is now time-varying. It was shown in [21, Section 5.3.2, p. 215] that the Hopf formula in (3.3.6) can be generalized for a time-varying Hamiltonian to find the value function of the system in (3.3.10) with

$$\varphi(z, t) = -\min_{p \in \mathbb{R}^n} \left\{ J_z^*(p, t) + \int_0^t H(p, s) ds - \langle z, p \rangle \right\}, \quad (3.3.11)$$

with the time-varying Hamiltonian defined as

$$H(p, t) = \max_{c \in \mathcal{C}} \langle e^{-(T-t)A}B(T-t)c, p \rangle.$$

The change of variable to $(T-t)$ is required for time since the problem was converted to an initial value formulation from a terminal value formulation in (3.3.4).

3.3.3 Linear Differential Games

Now consider the system

$$\dot{x}(t) = Ax(t) + B(t)u(t) + D(t)w(t), \quad (3.3.12)$$

with $D(t) \in \mathbb{R}^{n \times \ell}$, which is equal to (3.3.8) with an extra term, $D(t)w(t)$, added. We assume that the additional control input $w(t)$ is adversarial and bounded by $w(t) \in \mathcal{D} \subset \mathbb{R}^\ell$. The cost functional becomes

$$G(x, t, u, w) = \int_t^T L(u(t), w(t)) dt + J(x(T)), \quad (3.3.13)$$

where $x(T)$ is the solution of (3.3.12) at terminal time, T . We assume that the goal of the adversarial control input $w(t)$ is to *increase* the cost functional (3.3.13), in direct contradiction with the input $u(t)$, which we are designing in an attempt to minimize the cost. This system forms a differential game [40], and has a corresponding lower value function

$$V(x, t) = \inf_{u \in \mathcal{C}} \sup_{w \in \mathcal{D}} G(x, t, u, w),$$

and upper value function

$$U(x, t) = \sup_{w \in \mathcal{D}} \inf_{u \in \mathcal{C}} G(x, t, u, w).$$

As derived in [46], the upper and lower value functions are viscosity solutions of possibly non convex HJ equation. We can define the following upper and lower Hamiltonians as

$$H^+(p, t) = \sup_{c \in \mathbb{R}^m} \inf_{d \in \mathbb{R}^\ell} \{ \langle -f(t, c, d), p \rangle - L(c, d) \},$$

$$H^-(p, t) = \inf_{d \in \mathbb{R}^\ell} \sup_{c \in \mathbb{R}^m} \{ \langle -f(t, c, d), p \rangle - L(c, d) \}.$$

The running cost becomes

$$L(u, w) = \mathcal{I}_C(u) - \mathcal{I}_D(w),$$

where $\mathcal{I}_{\mathcal{D}}$ is the indicator function of the convex set \mathcal{D} . If the Hamiltonians H^+ and H^- coincide, then from [46]

$$H^+(p, t) = H^-(p, t) = H^\pm(p, t) \implies U(x, t) = V(x, t).$$

We can apply the same change of variables from (3.3.9) to get

$$\dot{z}(t) = e^{-tA}B(t)u(t) + e^{-tA}D(t)w(t), \quad (3.3.14)$$

and then we can find a candidate solution of the value function $\varphi(z, t) = U(z, t) = V(z, t)$ with the generalized Hopf formula

$$\varphi(z, t) = -\min_{p \in \mathbb{R}^n} \left\{ J_z^*(p, t) + \int_0^t H^\pm(p, s) ds - \langle z, p \rangle \right\},$$

with the time-varying, non convex Hamiltonian given by

$$\begin{aligned} H^\pm(p, t) &= \max_{c \in \mathcal{C}} \langle e^{-(T-t)A}B(T-t)c, p \rangle \\ &\quad - \max_{d \in \mathcal{D}} \langle e^{-(T-t)A}D(T-t)d, p \rangle. \end{aligned} \quad (3.3.15)$$

In general, if $H^+(p, t) \neq H^-(p, t)$, then the Hopf formula in (3.3.15) does not hold.

3.4 Time-Optimal Control with the Hopf Formula

Following the methods presented above in (3.3.14), we have the transformed system (3.2.4) as

$$\dot{z}(t) = e^{-t\hat{A}}\hat{B}a_p(t) + e^{-t\hat{A}}\hat{D}a_e(t),$$

and the Hamiltonian is the dual norm of the control set

$$H(p, t) = \left\| Q_p(T-t) \hat{B}^\dagger e^{-(T-t)\hat{A}^\dagger} p \right\|_1 - \left\| Q_e \hat{D}^\dagger e^{-(T-t)\hat{A}^\dagger} p \right\|_1, \quad (3.4.1)$$

where we denote by $\|(\cdot)\|_1$ the 1-norm. We choose a convex terminal cost function $J(z, 0)$ such that

$$\begin{cases} J(z, 0) < 0 & \text{for any } z \in \text{int } \Omega, \\ J(z, 0) > 0 & \text{for any } z \in (\mathbb{R}^n \setminus \Omega), \\ J(z, 0) = 0 & \text{for any } z \in (\Omega \setminus \text{int } \Omega), \end{cases} \quad (3.4.2)$$

where $\text{int } \Omega$ denotes the interior of Ω . The intuition behind defining the terminal cost function this way is simple. If the value function $\varphi(z_0, T) < 0$ for some z_0 and T , then there exists a control $u(t)$ that drives the state from the initial condition at z_0 , to the final state, $z(T)$ inside the set Ω . The smallest value of time T , such that $\varphi(z_0, T) = 0$ is the minimum time to reach the set Ω , starting at state z_0 . The control associated with the minimum time to reach is the time-optimal control. The ellipsoid terminal set defined in (3.2.3) results in a quadratic terminal cost function

$$J_x(x) = \langle x, W^{-1}x \rangle - 1,$$

After variable substitution the cost function becomes

$$J_z(z) = \langle z, V(0)z \rangle - 1,$$

with $V(t) = e^{(T-t)\hat{A}^\dagger} W^{-1} e^{(T-t)\hat{A}}$. Following the property that the Fenchel-Legendre transform of a norm function is the dual norm [47], we have

$$J_z^*(p, t) = 1 + \frac{1}{4} \langle p, V(0)^{-1} p \rangle.$$

The generalized Hopf formula requires the integration of the Hamiltonian which is approximated by Riemann sum quadrature [48] with step size h

$$\int_0^t H(p, t) ds \approx h \sum_{s_k \in \mathcal{S}} H(p, s_k),$$

where \mathcal{S} denotes the set of discrete time samples. Rectangular quadrature with fixed step size h was used to pre-compute the time samples s_k from time 0 to T , which requires only a simple sum at run time to evaluate the integral. We can approximate the matrix exponential terms efficiently at fixed time intervals, with bounded error, using [27].

To solve the Hopf formula in (3.3.11), we are performing an unconstrained minimization problem where the objective function is non-smooth. Non-smooth unconstrained minimization problems can be solved in a variety of ways. However, because we can explicitly derive the gradient and Hessian, this directs the use of a relaxed Newton's method [49]. We chose for the initial guess of Newton's method $p_0 = \frac{V(0)z}{2}$, the minimum of the Hopf objective without the Hamiltonian integral. The initial step size is 1 (full Newton), and is halved whenever the function value increases during an iteration (without updating the search direction). The minimization is terminated when the norm of the change in iterations is small. Most importantly for efficient implementation, the gradient and Hessian (ignoring discontinuities), denoted as ∇_p and \mathcal{H}_p , respectively, for

the minimization can be found directly. The gradient is

$$\begin{aligned}\nabla_p \varphi(z, t) &= \frac{V(0)^{-1} p}{2} - z \\ &+ h \sum_{s_k \in \mathcal{S}} \left(Q_p(s_k) E_p(s_k) \operatorname{sgn} \left(E_p(s_k)^\dagger p \right) \right. \\ &\left. - Q_e(s_k) E_e(s_k) \operatorname{sgn} \left(E_e(s_k)^\dagger p \right) \right),\end{aligned}$$

with $E_p(t) = e^{-(T-t)\hat{A}}\hat{B}$, and $E_e(t) = e^{-(T-t)\hat{A}}\hat{D}$. Additionally the Hessian is

$$\mathcal{H}_p(\varphi(z, t)) = \frac{V(0)^{-1}}{2}.$$

To find the optimal control to the desired convex terminal set Ω , we proceed by solving for the T^* , the minimum time to reach the boundary of the set Ω . This is solved numerically with

$$T^* = \arg \min_{t < T} \varphi(z_0, t).$$

If the minimum time to reach T^* is greater than total available time T , then the set Ω is not reachable in time T . The optimal control can then be found from the following relation

$$\nabla_p H(\nabla_z \varphi(z_0, T^*), T^*) = e^{-t\hat{A}}\hat{B}(t) a_p^* + e^{-t\hat{A}}\hat{D}(t) a_e^*.$$

To induce collaborative guidance we proceed to solve for the joint terminal set in (3.2.6). Let J_i represent terminal cost function of vehicle i with shape matrix W_i , then the terminal cost function of the collaborative system is

$$J(z, t) = \min_{i=1, \dots, k} J_i(z, t). \quad (3.4.3)$$

It was shown in [10] that max/min-plus algebra [50, 51, 52] can be used to generalize the Hopf formula to solve for non-convex initial data that can be formed as the union of convex sets, such as the terminal cost considered in (3.2.6). This is true provided that the Hamiltonian is convex. In general, the Hamiltonian of the differential game given in (3.4.1) is non-convex. But consider the case where $Q_e \leq Q_p$ and the system is constrained to the form in (3.2.2) and (3.2.4), then (3.4.1) is convex and max/min-plus algebra holds. To find the value function with the terminal set given by (3.4.3), we solve the k initial value problems of the form

$$\begin{cases} \frac{\partial \phi_i}{\partial t}(z, t) + H(t, \nabla_z \phi_i(z, t)) = 0 & \text{in } \mathbb{R}^n \times (0, +\infty), \\ \phi_i(z, 0) = J_i(z) & \forall z \in \mathbb{R}^n, \end{cases} \quad (3.4.4)$$

and take the pointwise minimum over the k solutions $\phi_i(z, t)$, each of which has convex initial data, with

$$\varphi(z, t) = \min_{i=1, \dots, k} \phi_i(z, t).$$

Each $\phi_i(z, t)$ in (3.4.4) are independent of each other, and can be computed in parallel. In the case where the (3.4.1) is non-convex, then the pointwise minimum is only an upper bound of the true value function; see [53] for more details.

3.5 Results

The above control solution has been integrated into a closed loop 2-on-1 pursuit-evasion 3 degree of freedom (3DOF) simulation using MATLAB R2016a and Simulink at 120Hz with Euler integration. This included using a third order autopilot for each pursuer, and using the gradient of the value function to find optimal evader control. Preliminary results solved for the optimal control on average 40 – 83 *ms* on a 3 GHz Intel

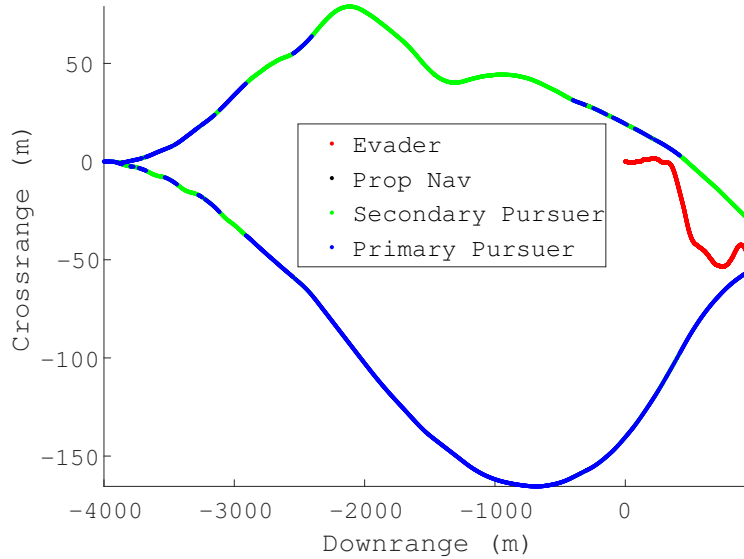


Figure 3.5.1: The trajectory of Example 1, a tail chase scenario. The red is the trajectory of the evader. The pursuing vehicles are shown in green and blue. Blue indicates at that time, it was the pointwise minimum of the k initial value problems in (3.4.4), while green was not.

Core i7 950.

As a post-process, the evader's inertial state is found by solving the modified Dubin's car initial-value problem (3.2.1) relative to a fixed origin with zero initial conditions and known inputs. Adding the evader's inertial state to the vehicle's relative state and correcting for the induced rotational motion provides the vehicle's inertial state.

The first example uses a simple geometry in the tail-chase scenario and the engagement trajectory is shown in Figure 3.5.1. The capture radius is $r = 3\text{ m}$, evader control is limited to $\|a_e\| \leq 10\text{ m/s}^2$, and both pursuers have control bounds that decrease in time with

$$\|a_p\| \leq \frac{(t - 40)^2}{40} \text{ m/s}^2,$$

when $0 \leq t \leq 40$, and 0 otherwise. The evader is assumed to travel at speed $V_e = 50\text{ m/s}$ and the pursuers at speed 255.225 m/s (0.75 Mach). Both pursuing vehicles, initially launched at 4000 m from the evader, are simultaneously traveling directly at the evader.

Notice that both pursuers separate as to surround and contain the evader. The miss distance was $0.879 m < r = 3 m$ and time to intercept was 19.533 seconds. In this example, $Q_e \leq Q_p$ and the Hamiltonian remained convex for the duration of the simulated engagement.

The second example utilized a similar engagement, but with head-on aspect configuration. The parameters are the same as example 1, but a $6000 m$ initial separation. In this case, the initial conditions are such that during simulation, the linearization error in (3.2.2) is large. When this occurs, the solution of the zero level set time maybe higher than available flight time T . This indicates the set Ω is not reachable (due to the linearization error) and in our simulations reverts to proportional navigation (PN) until the set Ω is considered reachable. This can easily be countered by increasing the control bound of the evader to account for linearization error. Additionally, the convexity assumption of $Q_e \leq Q_p$ is violated in this example, but only for the last 0.158 seconds, or about 0.78% of the engagement. With both vehicles launched simultaneously, intercept still occurred, with a miss distance of $2.34 m$. Time to intercept was 20.158 seconds and the flyout paths are given in Figure 3.5.2.

3.6 Conclusions and Future Work

The generalized Hopf formula provide new capabilities for solving high-dimensional optimal control and differential games, such as the pursuit-evasion guidance presented here. Additionally, the above work can be used for evasion strategies that could be of interest for collision avoidance problems. Future work will focus on extending the generalized Hopf formula for certain classes of non-linear systems, such as feedback linearizable systems [54], and apply splitting algorithms [10, 13, 23] for efficient optimization when the gradient and Hessian is not explicitly known.

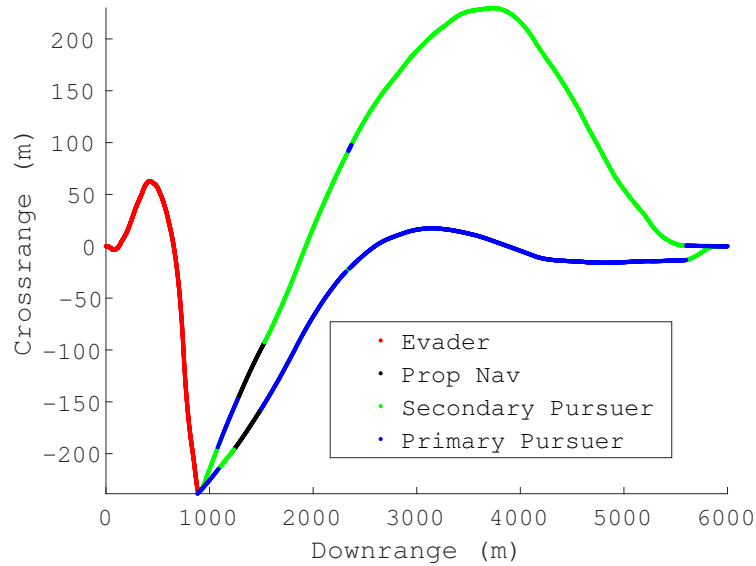


Figure 3.5.2: The trajectory of Example 2, a head-on scenario. The red is the trajectory of the evader. The pursuing vehicles are shown in green and blue. Blue indicates at that time, it was the pointwise minimum of the k initial value problems in (3.4.4), while green was not. Black is when Ω was considered not reachable due to high linearization error and proportional navigation was used.

Part II

Decomposition Approaches to Solving Hamilton–Jacobi Equations

Decomposition-based approaches to large scale systems is presented. In this family of techniques, we leverage the fact that some problems are too large to be effectively computed using existing methods but can be decomposed into smaller subproblems. Each of these subproblems are small enough to be efficiently computed using existing methods, and then the individual subproblem solutions can be aggregated to find the viscosity solution of the original problem. Decomposition methods are well suited for systems comprised of multiple vehicles collaborating toward a joint goal and one such technique is shown in Chapter 4. Motivated by a sensor selection problem given in Chapter 5, we create a hybrid method of lines approach by decomposing the state space into two parts: a low dimensional vehicle dynamics part, for which we construct a grid, and a high dimensional where no grid is needed.

Chapter 4

A Hamilton–Jacobi Formulation for Optimal Coordination of Heterogeneous Multiple Vehicle Systems

4.1 Introduction

Multi-robot motion planning presents several challenges, and key among them are trajectory planning and formation control, as well as the assignment of vehicles to goal states. When the assignment of vehicles to goal states is made beforehand, many methods have been proposed. These include control theory approaches [43, 55], graph-based techniques [56], and optimization methods [57, 58]. Approximations to assist in dimensionality reductions have also been proposed, such as sequential methods [59] and hybrid approaches [60, 61].

When the goal states of multi-vehicle teams are not set a priori, it is necessary to

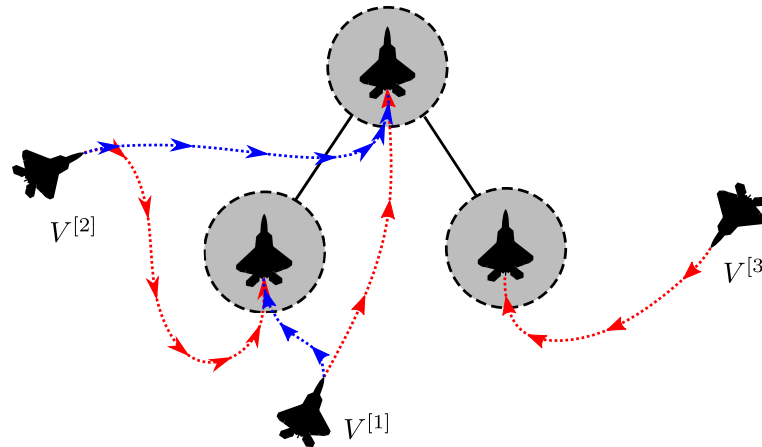


Figure 4.1.1: An illustration of the vehicle coordination problem. If a vehicle’s position in the formation is not assigned a priori, then many different configurations are possible to achieve the desired formation, each with a differing system objective cost. Two such possibilities are shown, one in red and an alternate assignment in blue.

determine which vehicle to allocate to which terminal goal. There is a large body of literature developing allocation and assignment algorithms [62], and applying these methods to engineering problems has appeared in diverse forms including sensor coverage [63], weapon-target assignment [64], or network routing [65]. Dynamics are not considered in this body of work, which commonly study worst-case system performance under equilibrium conditions using a game-theoretical framework [66, 67, 68]. Without consideration of dynamics, these methods have limited applicability to vehicle coordination problems and there exists a technical gap relating vehicle-goal allocation and motion planning, as most prior work assumes each is done independently. The critical dependency between assignment and trajectory planning is illustrated in Figure 4.1.1.

There have been recent attempts to close this gap as in [69]. But this work only evaluates discrete spatial waypoints on a rectangular grid without considering vehicle dynamics, and therefore the paths generated are not feasible for most physical systems. Turpin et al. [70] considered continuous dynamics, but was restricted to only single integrator dynamics and assumes homogeneous teams of vehicles. Morgan et al. [71]

generalizes this concept to linear, discrete-time dynamics and constructs a non-linear programming problem that is solved with sequential convex programming in conjunction with an auction algorithm [62]. However, both [71] and [69] restrict the cost function to one of additive individual vehicle weights (similar to the cost function presented in [62]), which excludes many vehicle optimization problems of interest, including assembling the desired formation shape in minimal time.

We present how the optimal assignment and trajectory can be found simultaneously from the viscosity solution to a single Hamilton–Jacobi PDE, a necessary and sufficient condition for optimality. We assume the vehicles can each have unique linear dynamics. Remarkably, we show that the global viscosity solution can be found as the solution to a linear bottleneck assignment problem (LBAP) [72]. This fact can be utilized to construct a level set method that has polynomial computational scaling with the number of vehicles and still ensures a global optimum.

Our formulation has a close relation to reachability analysis which can be useful for the design and analysis of heterogeneous systems. When the dynamics of some vehicles differ greatly from the rest, some formations may not be feasible (e.g. a slow-moving vehicle attempting to join a formation with a much faster vehicle). The zero sub-level set of the viscosity solution of a related HJ PDE defines an implicit surface representation of the backwards reachable set [7], and from this one determines whether the formation can be achieved given the unique collection of vehicle dynamics.

Traditionally, numerical solutions to HJ equations require a dense, discrete grid of the solution space [5, 6]. Computing the elements of this grid scales poorly with dimension and has limited use for problems with dimension greater than four. The exponential dimensional scaling in optimization is sometimes referred to as the “curse of dimensionality” [8, 9]. A new result in [10] discovered a numerical solution based on the Hopf formula [19] that does not require a grid and can be used to efficiently compute solu-

tions of a certain class of Hamilton–Jacobi PDEs. However, that only applies to systems with time-independent Hamiltonians of the form $\dot{x} = f(u(t))$, and has limited use for general linear control problems. Recently, the classes of systems were expanded upon and generalizations of the Hopf formula are used to solve optimal linear control problems in high-dimensions [73] and differential games as applied to multi-vehicle, collaborative pursuit-evasion problems [11].

The key contributions of this paper are to formulate the multi-vehicle coordination problem as a single Hamilton–Jacobi equation, show that the solution of which can be found from an LBAP, and use the generalized Hopf formula to simultaneously solve for optimal trajectory, vehicle control, and goal assignment. By utilizing the generalized Hopf formula, we create a level-set method to find the viscosity solution of the HJ in a computationally efficient manner.

We introduce a multi-vehicle dynamical system and formulate the coordination problem in Section 4.2. Section 4.3 formulates the optimal vehicle coordination problem as the viscosity solution to a single Hamilton–Jacobi PDE. Section 4.4 constructs a level-set method for the multi-vehicle coordination problem and utilizes the generalized Hopf formula as a computational tool to find the viscosity solution to the HJ PDE of the joint, multi-vehicle system. Section 4.5 shows the method as applied to several examples including a case of planar motion planning with 4 vehicles. This example shows how assignment and path planning cannot be decoupled and even if a single vehicle has a different initial condition, it may result in a different assignment for optimal coordination.

4.2 Problem Formulation

We consider a system that consists of N vehicles and each vehicle, $i \in \mathcal{V} = \{1, \dots, N\}$, has linear dynamics

$$\frac{d}{ds}x_i(s) := A_i x_i(s) + B_i \alpha_i(s), \quad (4.2.1)$$

for $s \in [0, t]$, where $x_i \in \mathbb{R}^{n_i}$ is the system state and $\alpha_i(s) \in \mathcal{A}_i \subset \mathbb{R}^{m_i}$ is the control input, constrained to a convex admissible control set \mathcal{A}_i . We add the assumption that no eigenvalue of matrix A_i has a strictly positive real part. We let $[0, t] \ni s \mapsto \gamma_i(s; x_i, \alpha_i(\cdot)) \in \mathbb{R}^{n_i}$ denote a state trajectory for vehicle i that evolves in time with measurable control sequence $\alpha_i(\cdot) \in \mathcal{A}_i$, according to (4.2.1) starting from initial state x_i at $s = 0$. The trajectory γ_i is a solution of (4.2.1) in that it satisfies (4.2.1) almost everywhere:

$$\begin{aligned} \frac{d}{ds}\gamma_i(s; x_i, \alpha_i(\cdot)) &= A_i \gamma_i(s; x_i, \alpha_i(\cdot)) + B_i \alpha_i(s), \\ \gamma_i(0; x_i, \alpha_i(\cdot)) &= x_i. \end{aligned}$$

4.2.1 Multi-Vehicle Model

For the set of N vehicles, we construct a joint state space with state vector $x = (x_1, \dots, x_i, \dots, x_N) \in \mathbb{R}^n$ and control $\alpha = (\alpha_1, \dots, \alpha_i, \dots, \alpha_N) \in \mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_i \times$

$\dots \times \mathcal{A}_N \subset \mathbb{R}^m$ which is written as follows

$$\begin{aligned} \dot{x} = \begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_i \\ \vdots \\ \dot{x}_N \end{bmatrix} &= \begin{bmatrix} A_1 & \cdots & 0 \\ & \ddots & \\ \vdots & A_i & \vdots \\ & & \ddots \\ 0 & \cdots & A_N \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_N \end{bmatrix} \\ &+ \begin{bmatrix} B_1 & \cdots & 0 \\ & \ddots & \\ \vdots & B_i & \vdots \\ & & \ddots \\ 0 & \cdots & B_N \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_i \\ \vdots \\ \alpha_N \end{bmatrix} \\ &\implies \dot{x} = Ax + B\alpha, \end{aligned} \tag{4.2.2}$$

We reiterate the above definition of the joint state space with the fact that when quantities x , A , B , γ , and α appear without subscripts, they refer to the joint system in (4.2.2), and when subscripts are used, they refer to a specific individual vehicle as defined in (4.2.1).

4.2.2 Vehicle Coordination

We assume there exists a set of N goals, and for each vehicle $i \in \mathcal{V}$ we associate closed convex sets $\Omega_{i,j} \subset \mathbb{R}^{n_i}$, $j \in \mathcal{V}$ with the understanding that $x_i \in \Omega_{i,j}$ means the vehicle i is assigned to goal j . Our goal is to make sure that we have one vehicle at each goal, but it does not matter which vehicle is at each goal. This goal can be expressed by the requirements that the multi-vehicle state, x , belongs to the following goal set

$$\Theta := \{x \in \mathbb{R}^n : \exists \sigma \in \mathcal{S}_N \text{ s.t. } \forall i \in \mathcal{V}, x_i \in \Omega_{i,\sigma(i)}\}, \tag{4.2.3}$$

where \mathcal{S}_N denotes the set of all permutations of \mathcal{V} . We represent Θ implicitly with the function $J : \mathbb{R}^n \rightarrow \mathbb{R}$ such that

$$\Theta = \{x \in \mathbb{R}^n \mid J(x) \leq 0\}, \quad (4.2.4)$$

and use it to construct a cost functional for the system trajectory $\gamma(s; x, \alpha(\cdot))$, given terminal time t as

$$R(t, x, \alpha(\cdot)) = \int_0^t C(\alpha(s)) ds + J(\gamma(t; x, \alpha(\cdot))), \quad (4.2.5)$$

where the function $C : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$ represents the rate that cost is accrued over time. The value function $\varphi : \mathbb{R}^n \times (0, +\infty) \rightarrow \mathbb{R}$ is defined as the minimum cost, R , among all admissible controls for a given initial state x with

$$\varphi(x, t) = \inf_{\alpha(\cdot) \in \mathcal{A}} R(t, x, \alpha(\cdot)). \quad (4.2.6)$$

4.3 Hamilton–Jacobi Equations For Optimal Coordination

The value function in (4.2.6) satisfies the dynamic programming principle [17, 18] and also satisfies the following initial value Hamilton–Jacobi (HJ) equation with φ being the viscosity solution of

$$\begin{aligned} \frac{\partial \varphi}{\partial s}(x, s) + H(s, x, \nabla_x \varphi(x, s)) &= 0, \\ \varphi(x, 0) &= J(x), \end{aligned} \quad (4.3.1)$$

for $s \in [0, t]$, where the Hamiltonian $H : (0, +\infty) \times \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is defined by

$$H(s, x, p) = -x^\top A^\top p + \sup_{\alpha \in \mathbb{R}^m} \{ \langle -B\alpha, p \rangle - C(\alpha) \}, \quad (4.3.2)$$

where $p := \nabla_x \varphi(x, s)$ in (4.3.2) denotes the *costate*. We denote by $[0, t] \ni s \mapsto \lambda(s; x, \alpha(\cdot)) \in \mathbb{R}^n$ the costate trajectory that can be shown to satisfy almost everywhere:

$$\begin{aligned} \frac{d}{ds} \lambda(s; x, \alpha(\cdot)) &= \nabla_x f(\gamma(s; x, \alpha(\cdot)), s)^\top \lambda(s; x, \alpha(\cdot)) \\ \lambda(t; x, \alpha(\cdot)) &= \nabla_x \varphi(\gamma(t; x, \alpha(\cdot))), \end{aligned}$$

with initial costate denoted by $\lambda(0; x, \alpha(\cdot)) = p_0$. With a slight abuse of notation, we will hereafter use $\lambda(s)$ to denote $\lambda(s; x, \alpha(\cdot))$, when the initial state and control sequence can be inferred through context with the corresponding state trajectory, $\gamma(s; x, \alpha(\cdot))$.

4.3.1 System Hamiltonian

We consider a time-optimal formulation with

$$C(\alpha) = \mathcal{I}_{\mathcal{A}}(\alpha),$$

where $\mathcal{I}_{\mathcal{A}} : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$ is the characteristic function of the set of admissible controls and is defined by

$$\mathcal{I}_{\mathcal{A}}(\alpha) = \begin{cases} 0 & \text{if } \alpha \in \mathcal{A} \\ +\infty & \text{otherwise.} \end{cases}$$

In this case, the integral term in (4.2.5) disappears (as long as α remains in \mathcal{A}) and $R(t, x, \alpha(\cdot))$ is simply the value of $J(\gamma(t; x, \alpha(\cdot)))$ at the terminal time, t . For this

problem

$$\varphi(x, t) = \inf_{\alpha(\cdot) \in \mathcal{A}} J(\gamma(t; x, \alpha(\cdot))) \leq 0$$

means that there exists a feasible trajectory for the vehicles that ends at the state $x(t) \in \Theta$; whereas $\varphi(x, t) > 0$ means that such trajectories do not exist under the system dynamics and initial conditions. Since each vehicle has independent dynamics, the Hamiltonian in (4.3.2) is of the form

$$H(s, x, p) = \sum_i H_i(s, x_i, p_i), \tag{4.3.3}$$

where each vehicle’s Hamiltonian is given by

$$\begin{aligned} H_i(s, x_i, p_i) = & -x_i^\top A_i^\top p_i \\ & + \sup_{\alpha_i \in \mathbb{R}^{m_i}} \{ \langle -B_i \alpha_i, p_i \rangle - \mathcal{I}_{\mathcal{A}_i}(\alpha_i) \}. \end{aligned} \tag{4.3.4}$$

4.3.2 Linear Bottleneck Assignment

Our goal is solving the Hamilton–Jacobi PDE in (4.3.1), and any $J(x)$ that satisfies (4.2.4) is, in general, non-convex and presents numerical challenges. We show that we can overcome this challenge by alternatively solving for the global value function with the following linear bottleneck assignment problem [72]:

$$\varphi(x, t) = \min_{\sigma \in \mathcal{S}_N} \max_{i \in \mathcal{V}} \phi_{i, \sigma(i)}(x_i, t), \tag{4.3.5}$$

where $\phi_{i,j}(x_i, t)$ is the viscosity solution to

$$\begin{aligned} \frac{\partial \phi_{i,j}}{\partial s}(x_i, s) + H_i(s, x, \nabla_x \phi_{i,j}(x_i, s)) &= 0, \\ \phi_{i,j}(x_i, 0) &= J_{i,j}(x_i), \end{aligned} \tag{4.3.6}$$

with Hamiltonian defined by (4.4.9). The function $J_{i,j} : \mathbb{R}^{n_i} \rightarrow \mathbb{R}$ is an implicit surface representation of $\Omega_{i,j}$ such that

$$\Omega_{i,j} = \{x_i \in \mathbb{R}^{n_i} | J_{i,j}(x_i) \leq 0\}. \quad (4.3.7)$$

The solution to (4.3.5) can be found from $\phi_{i,j}$ using the appropriate linear bottleneck assignment algorithms (for example [74, Section 6.2]), requiring only N^2 evaluations of lower dimensional viscosity solutions and avoiding computation involving all $|\mathcal{S}_N| = N!$ permutations of vehicle-goal pairs. Also of note is that if each HJ equation (4.3.6) has convex initial data, it enables the use of computationally efficient techniques that can guarantee convergence to the appropriate viscosity solution. Additionally, since each of the solutions are independent, the N^2 solutions can be computed in parallel.

We introduce a set of mild regularity assumptions that guarantee Hamilton–Jacobi equations have a unique viscosity solution [75, Chapter 7, p. 63]:

1. Each Hamiltonian

$$[0, t] \times \mathbb{R}^{n_i} \times \mathbb{R}^{n_i} \ni (s, x_i, p_i) \mapsto H_i(s, x_i, p_i) \in \mathbb{R}, \forall i$$

is continuous.

2. There exists a constant $c_i > 0$ such that for all $(s, x_i) \in [0, t] \times \mathbb{R}^{n_i}$ and for all $p', p'' \in \mathbb{R}^{n_i}$, the following inequalities hold

$$|H_i(s, x_i, p') - H_i(s, x_i, p'')| \leq \rho_i(x_i) \|p' - p''\|, \forall i$$

and

$$|H_i(s, x_i, 0)| \leq \rho_i(x_i), \forall i$$

with $\rho_i(x_i) = c_i(1 + \|x_i\|)$.

3. For any compact set $M \subset \mathbb{R}^{n_i}$ there exists a constant $\kappa_i(M) > 0$ such that for all $x', x'' \in M$ and for all $(s, p_i) \in [0, t] \times \mathbb{R}^{n_i}$ the inequality holds, $\forall i$

$$|H_i(s, x', p_i) - H_i(s, x'', p_i)| \leq \mu_i(p_i) \|x' - x''\|,$$

with $\mu_i(p_i) = \kappa_i(M)(1 + \|p_i\|)$.

4. Each terminal cost function

$$\mathbb{R}^{n_i} \ni x_i \mapsto J_{i,j}(x_i) \in \mathbb{R}, \forall i, j$$

is continuous.

Lemma 4.1. *If each vehicle Hamiltonian, $H_i(s, x_i, p_i)$, meets assumptions 1-3, then the Hamiltonian defined in (4.3.3) also meets assumptions 1-3.*

The proof is given in the appendix.

Theorem 4.1. *Under assumptions 1-4, $\phi_{i,j}$ is a unique viscosity solution to (4.3.6) for all i, j , and there exists a $J : \mathbb{R}^n \rightarrow \mathbb{R}$ that satisfies (4.2.4) such that with the Hamiltonian given by (4.3.3), (4.3.5) is a viscosity solution to (4.3.1).*

Proof. We will prove the theorem constructively by proposing a particular $J : \mathbb{R}^n \rightarrow \mathbb{R}$ given as

$$J(x) = \min_{\sigma \in \mathcal{S}_N} J_\sigma(x), \tag{4.3.8}$$

with

$$J_\sigma(x) := \max_{i \in \mathcal{V}} J_{i,\sigma(i)}(x_i), \tag{4.3.9}$$

where $J_{i,\sigma(i)}(x_i)$ is the implicit representation of $\Omega_{i,j}$, defined in (4.3.7), with $i, j = i, \sigma(i)$. We see that for any $\sigma \in \mathcal{S}_N$ when $x_i \in \Omega_{i,\sigma(i)}, \forall i \in \mathcal{V} \implies J_{i,\sigma(i)}(x_i) \leq 0, \forall i$ which implies $J_\sigma(x) \leq 0$. We also see if there exists an i such that $x_i \notin \Omega_{i,\sigma(i)}$, then $J_{i,\sigma(i)} > 0$ and implies $J_\sigma(x) > 0$. Therefore the J proposed in (4.3.8) satisfies (4.2.4) and is an implicit surface representation of the set Θ . Furthermore, since by assumption each $J_{i,j}(x_i)$ is continuous, then (4.3.9) is convex as the max of a finite number of continuous functions is also continuous. This implies (4.3.8) is continuous as the minimum of a finite number of continuous functions is continuous. Therefore (4.3.8) satisfies assumption 4.

From Lemma 4.1, the system Hamiltonian defined in (4.3.3) meets assumptions 1-3 and implies that (4.3.1) has a unique viscosity solution, denoted as $\varphi_{\text{HJ}}(x, t)$, when the initial cost function is given by (4.3.8) [75, Theorem 8.1, p. 70]. The uniqueness of the solution $\varphi_{\text{HJ}}(x, t)$ implies that the viscosity solution is equivalent to the value function in (4.2.6). It follows that

$$\varphi_{\text{HJ}}(x, t) = \inf_{\alpha(\cdot) \in \mathcal{A}} J(\gamma(t; x, \alpha(\cdot))), \quad (4.3.10)$$

since $C(\alpha) = 0$ when $\alpha \in \mathcal{A}$. Denoting by α^* as the control that optimizes (4.3.10), we have

$$\varphi_{\text{HJ}}(x, t) = J(\gamma(t; x, \alpha^*(\cdot))).$$

Substituting (4.3.9) and (4.3.8) we have

$$\varphi_{\text{HJ}}(x, t) = \min_{\sigma \in \mathcal{S}_N} \max_{i \in \mathcal{V}} J_{i,\sigma(i)}([\gamma(t; x, \alpha^*(\cdot))]_i), \quad (4.3.11)$$

where we use the notation $[\gamma(t; x, \alpha^*(\cdot))]_i$ to represent the i -th block of the vector $\gamma(t; x, \alpha^*(\cdot))$. Likewise (4.3.6) has a unique viscosity solution, $\phi_{i,j}$, for each i, j with

initial cost given (4.3.7), and as such

$$\begin{aligned} \phi_{i,\sigma(i)}(x_i, t) &= \inf_{\alpha_i(\cdot) \in \mathcal{A}_i} J_{i,\sigma(i)}(\gamma_i(t; x_i, \alpha_i(\cdot))) \\ &= J_{i,\sigma(i)}(\gamma_i(t; x_i, \alpha_i^*(\cdot))), \end{aligned} \tag{4.3.12}$$

for each i and with α_i^* denoting the control that optimizes (4.3.12). Note that if $J_{i,\sigma(i)}([\gamma(t; x, \alpha^*(\cdot))]_i) < J_{i,\sigma(i)}(\gamma_i(t; x_i, \alpha_i^*(\cdot)))$ it would contradict (4.3.12) that α_i^* is the optimal control. Also if $J_{i,\sigma(i)}([\gamma(t; x, \alpha^*(\cdot))]_i) > J_{i,\sigma(i)}(\gamma_i(t; x_i, \alpha_i^*(\cdot)))$, then it would contradict (4.3.10) that α^* is the optimal control of the entire system. Therefore, $J_{i,\sigma(i)}([\gamma(t; x, \alpha^*(\cdot))]_i) = J_{i,\sigma(i)}(\gamma_i(t; x_i, \alpha_i^*(\cdot))) = \phi_{i,\sigma(i)}(x_i, t)$ and our value function in (4.3.10) becomes

$$\varphi_{\text{HJ}}(x, t) = \min_{\sigma \in \mathcal{S}_N} \max_{i \in \mathcal{V}} \phi_{i,\sigma(i)}(x_i, t),$$

and we arrive at our result. □

4.4 A Level Set Method with the Generalized Hopf Formula

First, we introduce the Fenchel–Legendre transform of a convex, proper, lower semi-continuous function $g : \mathbb{R}^\ell \rightarrow \mathbb{R} \cup \{+\infty\}$, denoted as $g^* : \mathbb{R}^\ell \rightarrow \mathbb{R} \cup \{+\infty\}$, and is defined as [20]

$$g^*(p) = \sup_{y \in \mathbb{R}^\ell} \{\langle p, y \rangle - g(y)\}. \tag{4.4.1}$$

We propose to find the viscosity solutions of (4.3.6) using the generalized Hopf formula.

Theorem 4.2. *Each viscosity solution of (4.3.6) can be found from the formula*

$$\begin{aligned} \phi_{i,j}(x_i, t) = & - \min_{p_i \in \mathbb{R}^{n_i}} \left\{ J_{i,j}^* \left(e^{-tA_i^\top} p_i \right), \right. \\ & \left. + \int_0^t \widehat{H}_i(s, p_i) ds - \langle x_i, p_i \rangle \right\} \end{aligned} \quad (4.4.2)$$

with

$$\widehat{H}_i(s, p_i) = \sup_{\alpha \in \mathcal{A}_i} \left\{ \langle -e^{-sA_i} B_i \alpha_i, p_i \rangle \right\}. \quad (4.4.3)$$

Proof. Proceeding similar to [11], we apply a change of variables to (4.2.1) with

$$z_i(s) = e^{-sA_i} x_i(s), \quad (4.4.4)$$

which results in the following system

$$\frac{d}{ds} z_i(s) = e^{-sA_i} B_i \alpha_i(s). \quad (4.4.5)$$

By utilizing this change of variables, we construct an equivalent HJ equation

$$\begin{aligned} \frac{\partial \tilde{\phi}_{i,j}}{\partial s}(z_i, s) + \widehat{H}_i \left(s, \nabla_z \tilde{\phi}_{i,j}(z_i, s) \right) &= 0, \\ \tilde{\phi}_{i,j}(z, 0) &= J_{i,j}(e^{tA} z_i). \end{aligned} \quad (4.4.6)$$

From Lemma A.1 (given in the appendix), (4.4.3) meets assumption 1-3, and by composition rule $J_{i,j}(e^{tA} z_i)$ is continuous [76, Theorem 4.7] and meets assumption 4. Therefore (4.4.6) has a unique viscosity solution that is equivalent to the cost functional

$$\tilde{\phi}_{i,j}(z_i, t) = \inf_{\alpha_i(\cdot) \in \mathcal{A}_i} J_{i,j} \left(e^{tA} \xi_i(t; z_i, \alpha_i(\cdot)) \right),$$

where $\xi_i(s; z_i, \alpha(\cdot))$ is a solution trajectory that satisfies (4.4.5) almost everywhere. Since

$$z_i(t) = e^{-tA_i} x_i(t),$$

we have

$$\begin{aligned} \tilde{\phi}_{i,j}(z_i, t) &= \inf_{\alpha_i(\cdot) \in \mathcal{A}_i} J_{i,j}(e^{tA_i} \xi_i(t; z_i, \alpha_i(\cdot))) \\ &= \inf_{\alpha_i(\cdot) \in \mathcal{A}_i} J_{i,j}(\gamma_i(t; x_i, \alpha_i(\cdot))) \\ &= \phi_{i,j}(x_i, t), \end{aligned}$$

noting that since $x_i = \gamma_i(0; x_i, \alpha_i(\cdot))$ by the transform (4.4.4) implies $z_i = x_i$ at $t = 0$. Thus, $\tilde{\phi}(z_i, t) = \phi(x_i, t)$ and we can find $\phi(x_i, t)$ by finding the viscosity solution to (4.4.6).

Since $\Omega_{i,j}$ is assumed closed and convex, this implies $J_{i,j}$ is convex and by assumption 4 is continuous in z_i . By assumption 1, $H(p_i)$ is continuous in p_i , therefore (4.4.2) gives an exact, point-wise viscosity solution for (4.4.6) [21, Section 5.3.2, p. 215]. \square

Formula (4.4.2) shows that we can compute a viscosity solution to (4.4.6) by solving a finite dimensional optimization problem. This avoids constructing a discrete spatial grid, and is numerically efficient to compute even when the dimension of the state space is large. Additionally, no spatial derivative approximations are needed with Hopf formula-based methods, and this eliminates the numeric dissipation introduced with the Lax–Friedrichs scheme [77], which is necessary to maintain numeric stability in grid-based methods.

4.4.1 Numerical Optimization of the Hopf Formula

We transcribe the Hopf formula into a non-linear programming problem by approximating the integral in (4.4.2) with an N -point quadrature rule sampled on the time grid

$$\pi^N = \{s_k : k = 0, \dots, N\},$$

with $s_k \in [0, t]$ and corresponding weights w_k . Additionally, we make a simple, invertible change of variable $\tilde{p}_i = e^{-tA_i^\top} p_i$ and substituting into (4.4.2) results in the following unconstrained optimization problem that solves (4.4.2):

$$\left\{ \min_{\tilde{p}_i} J_{i,j}^*(\tilde{p}_i) + \sum_{j=0}^N w_j \widehat{H}_i(s_k, \tilde{p}_i) - \langle e^{tA_i} x_i, \tilde{p}_i \rangle, \right. \quad (4.4.7)$$

with \widehat{H}_i now being defined as

$$\widehat{H}_i(s, \tilde{p}_i) = \sup_{\alpha_i \in \mathcal{A}_i} \left\{ \langle -B_i \alpha_i, e^{sA_i^\top} \tilde{p}_i \rangle \right\}. \quad (4.4.8)$$

The variable transformation is done so that when the matrix A has at least one eigenvalue with strictly negative real part, we avoid computation of e^{-sA^\top} which would have exponentially unstable poles since we are evaluating the matrix exponential of $-A$. This divergence would cause sensitivity in the evaluation of the Hopf formula in (4.4.2) with respect to small changes of p . By utilizing the variable transformation and optimizing (4.4.7), we avoid this sensitivity.

Remark 4.1. Often the expression in (4.4.8) is known in closed form and we present one such common case. Recall that $(\cdot)^*$ denotes the Fenchel–Legendre transform of a function defined in (4.4.1), and suppose \mathcal{A}_i is the closed convex set defined as

$$\mathcal{A}_i := \{u : \|u\| \leq 1\},$$

where $\|(\cdot)\|$ is any norm. Then $(\mathcal{I}_{\mathcal{A}_i})^*$ defines a norm, which we denote with $\|(\cdot)\|_*$, which is the dual norm to $\|(\cdot)\|$ [20]. From this we write (4.4.8) as

$$\widehat{H}_i(s, \tilde{p}_i) = \left\| -B_i^\top e^{sA_i^\top} \tilde{p}_i \right\|_* . \quad (4.4.9)$$

4.4.2 Time-Optimal Control to a Goal Set

The task of determining the control that drives the system into Θ in minimal time can be determined by finding the smallest t such that

$$\varphi(x, t) \leq 0. \quad (4.4.10)$$

When the system (4.2.1) is constrained controllable, then the set of times such that x is reachable with respect to Θ contains the open interval $[t', \infty)$ [78]. This insures that if x is outside the set Θ at time $t = 0$, then $\varphi(x, 0) > 0$ and there exists a time t' such that $\varphi(x, \tau) < 0$ for all $\tau > t'$. This implies standard root-finding algorithms can be used to find (4.4.10). As noted in [73], we solve for the minimum time to reach the set Θ by constructing a newton iteration, starting from an initial guess, t_0 , with

$$t_{k+1} = t_k - \frac{\varphi(x, t_k)}{\frac{\partial \varphi}{\partial t}(x, t_k)}, \quad (4.4.11)$$

where $\varphi(x, t_k)$ is the solution to (4.3.1) at time t_k . The value function must satisfy the HJ equation

$$\frac{\partial \varphi}{\partial t}(x, t_k) = -H(s, x, \nabla_x \varphi(x, t_k)),$$

where $\nabla_x \varphi(x, t_k) = \left(e^{t_k A_1^\top} \tilde{p}_1^*, \dots, e^{t_k A_N^\top} \tilde{p}_N^* \right)$ and each \tilde{p}^* is the argument of the minimizer in (4.4.7). We iterate (4.4.11) until convergence at the optimal time to reach, which we denote as t^* . This process is summarized in Algorithm 1.

The optimal control and trajectory for each vehicle is found directly from the necessary conditions of optimality established by Pontryagin’s principal [79] by noting the optimal trajectory must satisfy

$$\begin{aligned} \frac{d}{ds} \gamma_i^* (s; x_i, \alpha_i^* (\cdot)) &= -\nabla_p H (s, x_i (s), \lambda_i^* (s)) \\ &= A_i \gamma_i (s; x_i, \alpha_i^* (\cdot)) \\ &\quad + B_i \nabla_p \left\| -B_i^\top \lambda_i^* (s) \right\|_* , \end{aligned}$$

where λ_i^* is the optimal costate trajectory and is given by

$$\lambda_i^* (s) = e^{-(t^*-s)A_i^\top} \tilde{p}_i^* .$$

This implies our optimal control is

$$\alpha_i^* (s) = \nabla_p \left\| -B_i^\top e^{-(t^*-s)A_i^\top} \tilde{p}_i^* \right\|_* , \tag{4.4.12}$$

for all time $s \in [0, t^*]$, provided the gradient exists.

4.5 Results

We present results for two examples. For the first we choose a toy example of two vehicles, each with differing single integrator dynamics. The purpose of this example is that it allows easy visualization of the level set propagation and the solution can be verified by comparing with methods such as [32], which cannot be applied to the higher dimensional second example to follow. The LBAP was solved using [74, Algorithm 6.1] and the optimization (4.4.7) used sequential quadratic programming [80, Chapter 18] with initial conditions $\tilde{p}_i = e^{tA_i} x_i (0)$ for each vehicle. To prevent a singular control

Algorithm 1 Algorithm to compute the viscosity solution at the minimum time to reach of the set Θ .

```

1: Inputs:
    $x_i, J_{i,j}, \forall i, j$ 
2: Initialize:
    $t = t_0, \varphi = \infty, \epsilon = 10^{-5}$ 
3: while  $|\varphi| \geq \epsilon$  do
4:   for all  $i, j$  do
5:      $Q_{i,j} = \min_{\tilde{p}_i} J_{i,j}^*(\tilde{p}_i) + \sum_{j=0}^N w_j \widehat{H}_i(s_k, \tilde{p}_i) - \langle e^{tA_i} x_i, \tilde{p}_i \rangle$ 
6:   end for
7:    $\varphi = \text{LBAP}(Q)$ 
8:    $p = \left( e^{tA_1^\top} \tilde{p}_1^*, \dots, e^{tA_i^\top} \tilde{p}_i^*, \dots, e^{tA_N^\top} \tilde{p}_N^* \right)$ 
9:    $t = t + \frac{\varphi}{H(x,p)}$ 
10: end while
11: Return:
    $p, t$ 

```

condition, a slight smoothing was applied to the Hamiltonian in (4.4.12) and (4.4.8) using [80] with parameter $\mu = 10^{-6}$.

4.5.1 Toy Problem

We present a two vehicle problem where the dynamics are $\dot{x}_1 = 3\alpha_1$ for the first vehicle and $\dot{x}_2 = \alpha_2$ for the second vehicle. The states x_1, x_2 are the linear position of each respective vehicle. The control is bounded by $|\alpha_i| \leq 1$. The goal states are $|x_i - 3| \leq 1$ for $j = 1$ and $|x_i + 3| \leq 1$ for $j = 2$. The level set contours of the joint space are shown in Figure 4.5.1, for ten time samples equally spaced on $t \in [0, 4]$. Notice this seemingly simple heterogeneous system can give rise to counterexamples for the assignment formulations given in [71] where the sum of the vehicle distances are used as the assignment metric and [69] where the time of arrival of vehicles was used. Take the initial state $x(0) = (4.667, 0.5)^\top$, which is shown on Figure 4.5.1. For the time metric, the assignment (1, 2) gives $0.222 + 2.5 = 2.7223$, while the assignment (2, 1) gives

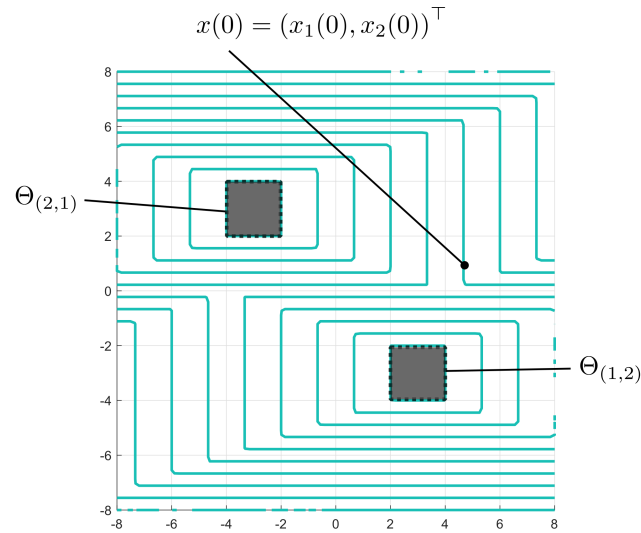


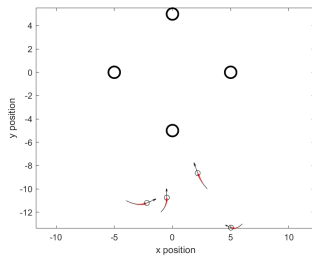
Figure 4.5.1: The two vehicle example of Section 4.5.1. The zero level set evolution solved at various times shown in green.

$2.222 + 1.5 = 3.722$, indicating $(1, 2)$ is the clear choice. However, $(2, 1)$ is the global optimum in this example, as both vehicles reach their desired goal states in a time of 2.222 , as the zero level set of the global value function intersects the point $x(0)$ at that time, giving the minimum time-to-reach. For the distance metric, the misassignment is more pronounced, with the assignment $(1, 2)$ giving $0.667 + 2.5 = 3.1670$ as compared to the assignment $(2, 1)$ gives $4.667 + 1.5 = 8.167$.

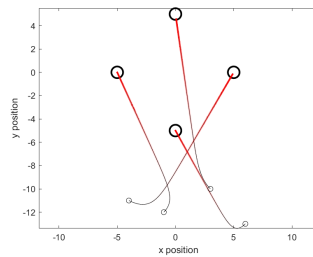
4.5.2 Planar Motion

We choose for dynamics (4.2.1) with state $x \in [r, \dot{r}]^\top$, where $r \in \mathbb{R}^2$ is spatial position of a robot and $\dot{r} \in \mathbb{R}^2$ is the velocity and

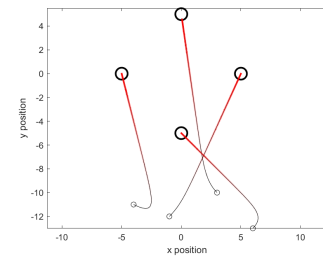
$$A_i = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, B_i = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$



(a) Optimal trajectories at $t = 1.50$ (10% of path).



(b) Complete optimal trajectories.



(c) Complete optimal trajectories for an alternate initial condition.

Figure 4.5.2: The example presented in Section 4.5.2 with four vehicles guiding to four possible goal sets. The optimal formation is achieved in $t^* = 15.015$. The trajectories are shown for two different times as they traverse the optimal path in the left two figures. The figure on the right is the trajectories when the initial condition for vehicle 4 is different.

for each vehicle $i \in \mathcal{V}$. The control $\alpha_i \in \mathbb{R}^2$ is constrained to lie in the set $\|\alpha_i\|_2 \leq 1$. The robots are tasked with reaching the goal formation and coming to rest, in minimum time. The goal sets each have radius of 0.5 and the centers of the goals are located spatially at $(0, 5)^\top$, $(-5, 0)^\top$, $(5, 0)^\top$, and $(0, -5)^\top$. Since the 2-norm is self-dual, the Hamiltonian (4.4.9) for each vehicle is

$$\hat{H}_i(s, \tilde{p}_i) = \left\| -B_i^\top e^{sA_i^\top} \tilde{p}_i \right\|_2.$$

Since A_i contains eigenvalues with negative real part, we optimize using the variable transformation of Section 4.4.1. The initial conditions of the vehicles were

$$\begin{aligned} x_1(0) &= (3, -10, -1, 1)^\top, \\ x_2(0) &= (-1, -12, 1, 1)^\top, \\ x_3(0) &= (-4, -11, 2, -1)^\top, \\ x_4(0) &= (6, -13, -1, -1)^\top. \end{aligned}$$

Only 11 iterations of (4.4.11) were needed to solve for the minimum time to reach the goal formation, which was found to be $t^* = 15.015$. Figures 4.5.2a and 4.5.2b show the optimal paths found from (4.4.12). Figure 4.5.2c shows the optimal paths when the initial condition for vehicle 4 was changed to $x_4(0) = (6, -13, 1, 1)^\top$ and a different optimal assignment results.

4.6 Conclusions and Future Work

We presented how to formulate vehicle coordination problems with unknown goal assignments as the viscosity solution to a single Hamilton–Jacobi equation. We show the solution of this single HJ PDE is equivalent to decomposing the problem and performing a linear bottleneck assignment using the viscosity solutions of independent single-vehicle problems. This allows quadratic computational scaling in the number of vehicles. Finally, a level set method based on the Hopf formula was presented for efficient computation of the vehicle value functions, in which each can be computed in parallel. Generalizations of the Hopf formula can be used in this context, for example [81]. The Hamilton–Jacobi formulation presented has other advantages for multi-robot systems, such as the compensation of time delays which can be induced in several ways including computation, sensing, and inter-robot communication. See for example [82]. Future work includes to expand the class of allowable vehicle dynamics from general linear models to certain types of nonlinear dynamics and to allow for dependencies between vehicles both in the dynamics models and in the cost functional.

Chapter 5

Heterogeneous Measurement Selection for Vehicle Tracking using Submodular Optimization

5.1 Introduction

The scenario we consider is that of a group of agents, each with multiple sensors collecting noisy measurements of a vehicle, and the measurements are transmitted over a communication channel to a centralized node. The central node collects the measurements and estimates a vector of unknown parameters that describes the motion of the vehicle. The communication channel restricts the amount of measurements that can be transmitted to the centralized node. Inspired by the Cramer-Rao lower bound for the error estimate, we propose to select measurements based on the Fisher information matrix (FIM), as “minimizing” the inverse of the FIM is required to achieve small estimation error.

One can use the FIM as a criteria to select which subset of measurements are “best”

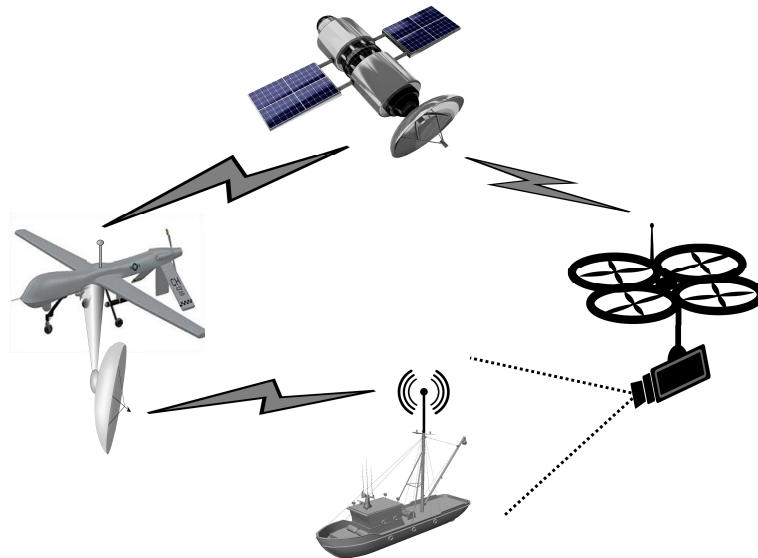


Figure 5.1.1: An example scenario where two agents separately take measurements of a ship and transmit them to a satellite for estimation. The ship being tracked has a radio transmitter and the agent on the left passively receives the signal. The second agent on the right is observing the ship with a camera.

by formulating a combinatorial optimization problem. However, this presents a computational challenge as finding the optimal selection of measurements is, in general, NP-hard. We show that one common criteria used to “minimize” the inverse of the FIM, maximizing $\log \det(\text{FIM})$, is both monotone and submodular and therefore allows the use of a greedy algorithm [83, Chapter 16] to find the selection of measurements. While the greedy algorithm returns a sub-optimal solution, it is guaranteed to be within $1 - 1/e \approx 63\%$ of the optimum and has the critical benefit of quadratic computational complexity.

There have been numerous proposals [84, 85, 86, 87] to use submodular optimization for sensor selection, however, these typically seek to optimize a criteria based directly on the estimated error covariance. As a result, they require simplified estimation models such as linear Kalman filtering to be used as in [84, 85] or Gaussian process regression on a fixed, discrete grid of points as in [86, 87]. In general, estimation problems involving vehicle tracking contain non-linear dynamics and sensor models that result in non-Gaussian

and often non-unimodal distributions, even when all the observation noises are simple independent zero-mean additive Gaussian distributions.

A key advantage of using the FIM is that we can utilize relatively simple and well-described distributions of the measurements, without having to know the (possibly complicated) distributions of the estimation error. This allows us to decompose the problem into two independent parts, one of measurement selection and another of performing estimates based on the selected measurements. The estimation can proceed with advanced estimation schemes such as non-parametric methods like particle filtering [36, Ch. 4.3, pp. 96] or optimization approaches [88], among others, without regard to the measurement selection process.

To illustrate this approach, we derive the FIM for different sensor types to which we apply measurement selection. This includes the time-of-arrival and Doppler shift of passively received radio transmissions as well as detected key-points in camera images. We compare the track estimation of the vehicle with the FIM selected measurements with that of random selection and show that selecting measurements based on the FIM can greatly outperform the estimation task when the bandwidth limitation becomes significant.

5.1.1 Problem Formulation

Consider an heterogeneous group of n mobile agents, each $i \in \{1, 2, \dots, n\}$ with a sensor that collects a set of m_i measurements, which we denote by

$$\{y_k^i : k \in \mathcal{K}_i\},$$

with $\mathcal{K}_i := \{1, 2, \dots, m_i\}$. Based on these measurements, we want to estimate a random variable, θ , of interest at a *measurement fusion center*. Due to bandwidth limitations,

each agent must select a subset of their own measurements to be transmitted to the remote sensor fusion center. We denote by $F_i \subset \mathcal{K}_i$ the indices of the measures that agent i sends to the fusion center. The bandwidth of the wireless channel imposes a constraint that $|F_i| \leq B_i$, where B_i is the maximum number of measurements that can be transmitted over the channel. The set of all measurements available to the fusion center is given by

$$\mathcal{Y}_{\text{fusion}} = \bigcup_{i=1}^n \{y_k^i : k \in F_i\}.$$

Our goal is to select the sets F_i such that this set of measurements contains the “best” $\sum_i B_i$ measurements from the perspective of estimating θ . Our goal is thus to design algorithms for each agent so they select subsets F_i that optimize

$$\min_{F_i \subset \mathcal{K}_i} \{f(F_i) : |F_i| \leq B_i\}, \quad (5.1.1)$$

where $f(F_i)$ is a metric that relates the selected measurements to estimation performance. We propose to use the Fisher information matrix (FIM) to construct the functions f as such to select the best measurements.

Directly finding the optimal value of (5.1.1) is computationally challenging and, in general, NP-hard [89]. This leads to approximations and heuristics to efficiently compute the selection, such as branch and bound [90] or convex relaxation [91]. Branch and bound can still be unreasonably slow while convex relaxations improve speed but is still cubic in complexity. Neither of these two methods provide any guarantee on the performance the approximate value relative to the true optimal.

5.2 Fisher Information Matrix

Assuming that all measurements y_k are conditionally independent given the unknown random variable, θ , the Bayesian Fisher Information Matrix (FIM) associated with the estimation of θ is given by

$$\text{FIM}(F) := Q_0 + \sum_{k \in F} Q_k \quad (5.2.1)$$

where

$$F := \bigcup_{i=1}^n F_i$$

denotes the set of all measurements sent to the fusion center,

$$Q_0 := \mathbb{E} \left[\frac{\partial \log p(\theta)}{\partial \theta} \frac{\partial \log p(\theta)}{\partial \theta}^\top \right], \quad (5.2.2)$$

denotes the contribution to the FIM due to the a-priori probability density function (pdf) $p(\theta)$ of θ , and

$$Q_k := \mathbb{E} \left[\frac{\partial \log p(y_k|\theta)}{\partial \theta} \frac{\partial \log p(y_k|\theta)}{\partial \theta}^\top \right], \quad (5.2.3)$$

the contribution to the FIM due to the measurement y_k with conditional pdf $p(y_k|\theta)$ given θ . In both (5.2.2) and (5.2.3), $\frac{\partial(\cdot)}{\partial \theta}$ denotes the gradient (as a row vector) with respect to the vector θ .

The relevance of the FIM to our problem stems from the (Bayesian) Cramér-Rao lower bound, which under the usual regularity assumptions on the pdfs gives

$$\mathbb{E} \left[\left(\theta - \hat{\theta} \right) \left(\theta - \hat{\theta} \right)^\top \right] \geq \text{FIM}(F)^{-1},$$

[92], which conceptually means that “minimizing” the inverse of the FIM is required to

achieve a small estimation error. In this paper, we propose to minimize the criteria

$$\begin{aligned} & \log \frac{\det(\text{FIM}(F)^{-1})}{\det(\text{FIM}(\emptyset)^{-1})} \\ &= -\log \det(\text{FIM}(F)) + \log \det(\text{FIM}(\emptyset)) \end{aligned} \quad (5.2.4)$$

which essentially corresponds to minimize the volume of the error ellipsoid, normalized by the volume of the a-priori error ellipsoid. Alternative criteria include

$$\frac{\text{trace}(\text{FIM}(F)^{-1})}{\text{trace}(\text{FIM}(\emptyset)^{-1})} \quad (5.2.5)$$

or

$$\frac{\lambda_{\max}(\text{FIM}(F)^{-1})}{\lambda_{\max}(\text{FIM}(\emptyset)^{-1})}, \quad (5.2.6)$$

which correspond to minimizing the achievable normalized mean-square estimation error $\mathbb{E}[|\theta - \hat{\theta}|^2]$ or the length of the largest axis of the error ellipsoid, respectively. However, we shall see shortly that (5.2.4) has the desirable property that it leads to a submodular optimization when the a-priori contribution to the FIM is nonsingular, whereas (5.2.5) and (5.2.6) do not share this property [93, 85].

5.3 FIM-Based Measurement Selection

The selection of a set of measurements by agent $i \in \{1, 2, \dots, n\}$ that minimizes the (normalized) volume of the error ellipsoid associated with its own measurements, subject

to communication constraints, can be formalized as the following maximization:

$$\begin{cases} \text{maximize} & f(F_i) \\ \text{subject to} & F_i \subset \mathcal{K}_i, \\ & |F_i| \leq B_i, \end{cases} \quad (5.3.1)$$

where $f(F_i)$ is the symmetric of (5.2.4) with FIM(F_i) given by (5.2.1) with $F = F_i$, which leads to

$$f(F_i) := \log \det \left(Q_0 + \sum_{k \in F_i} Q_k \right) - \log \det(Q_0). \quad (5.3.2)$$

We recall that a scalar-valued function $f : 2^{\mathcal{K}} \rightarrow \mathbb{R}$ that maps subsets of a finite set \mathcal{K} to \mathbb{R} is called *submodular* if for every $X \subset Y \subset \mathcal{K}$, $s \in \mathcal{K} \setminus Y$

$$f(X \cup \{s\}) - f(X) \geq f(Y \cup \{s\}) - f(Y), \quad (5.3.3)$$

monotone if

$$X \subseteq Y \implies f(X) \leq f(Y), \quad (5.3.4)$$

and *normalized* if

$$f(\emptyset) = 0. \quad (5.3.5)$$

Submodular functions are important for us because of the following well-known result in combinatorial optimization, which provides an algorithm to approximate the solution to (5.3.1) that has only quadratic complexity on the number of measurements and provides formal bounds on the performance of the approximation.

Theorem 5.1. [94] *When $f(\cdot)$ is normalized, monotone and submodular, then the Algorithm 2 returns a set F_i^* that leads to a criteria $f(F_i^*)$ no less than $1 - 1/e$ of the optimum of (5.3.1).*

Algorithm 2 Greedy Optimization Algorithm.

1: **Inputs:**
 $\{\Sigma_k\}_{k \in \mathcal{K}_i}, \{\mu_k\}_{k \in \mathcal{K}_i}$
 2: **Initialize:**
 $Q = Q_0, F_i = \emptyset, c = 0$
 3: **while** $c < B_i$ **do**
 4: $j = \arg \min_{k \in \mathcal{K}_i \setminus F_i} f(F_i \cup \{k\}) - f(F_i)$
 5: $Q = Q + \text{FIM}(j)$
 6: $F_i = F_i \cup j$
 7: $c = c + 1$
 8: **end while**
 9: **Return:**
 F_i

It turns out that the maximization in (5.3.1) has submodular structure:

Theorem 5.2. *Assuming that the a-priori pdf $p(\theta)$ leads to a positive definite matrix Q_0 in (5.2.2), the function $f(\cdot)$ in (5.3.2) is normalized, monotone, and submodular.*

To prove Theorem 5.2, we introduce a result on general functions f of the form

$$f(X) = g\left(Q_0 + \sum_{k \in X} Q_k\right), \forall S \subseteq \mathcal{K}, \quad (5.3.6)$$

where all $Q_0, Q_k, k \in \mathcal{S}$ are $n \times n$ matrices and g is a function from $\mathbb{R}^{n \times n}$ to \mathbb{R} . In the sequel, we use the notation

$$D_g(A) := \left[\frac{\partial g(A)}{\partial a_{ij}} \right]_{ij} \in \mathbb{R}^{n \times n},$$

which allow us to write

$$\begin{aligned} \frac{dg(A(\lambda))}{d\lambda} &= \sum_{i=1}^n \sum_{j=1}^n [D_g(A(\lambda))]_{ij} \frac{d[A(\lambda)]_{ij}}{d\lambda} \\ &= \text{trace} \left[D_g(A(\lambda))^\top \frac{dA(\lambda)}{d\lambda} \right]. \end{aligned} \quad (5.3.7)$$

Lemma 5.1. *Assume that Q_0 is a symmetric positive definite matrix, that all the Q_k , $k \in \mathcal{S}$ are symmetric positive semidefinite matrices, and that the function $g : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$ has the property that for every pair of symmetric positive definite matrices A, B , we have that $D_g(A)$ and $D_g(B)$ are both symmetric positive semidefinite and*

$$A \succeq B \implies D_g(A) \preceq D_g(B), \quad (5.3.8)$$

then the function f defined by (5.3.6) is monotone and submodular.

Proof. To prove that f is monotone, pick $X \subset Y \subset \mathcal{S}$ and define

$$\delta(\lambda) := g \left(Q_0 + \sum_{k \in X} Q_k + \lambda \sum_{k \in Y \setminus X} Q_k \right),$$

$\forall \lambda \in [0, 1]$. For this function we have that

$$\delta(0) = g \left(Q_0 + \sum_{k \in X} Q_k \right) = f(X),$$

and

$$\delta(1) = g \left(Q_0 + \sum_{k \in X} Q_k + \sum_{k \in Y \setminus X} Q_k \right) = f(Y),$$

and, in view of (5.3.7),

$$\begin{aligned} \frac{d\delta(\lambda)}{d\lambda} &= \text{trace} \left[D_g \left(Q_0 + \sum_{k \in X} Q_k + \lambda \sum_{k \in Y \setminus X} Q_k \right)^\top \right. \\ &\quad \left. \times \left(\sum_{k \in Y \setminus X} Q_k \right) \right]. \end{aligned} \quad (5.3.9)$$

Since Q_0 is positive definite, $Q_0 + \sum_{k \in X} Q_k + \lambda \sum_{k \in Y \setminus X} Q_k$ is also positive definite and

by assumption

$$D_g \left(Q_0 + \sum_{k \in X} Q_k + \lambda \sum_{k \in Y \setminus X} Q_k \right) \succeq 0.$$

Moreover, because the trace of the product of two positive semidefinite matrices is non-negative, we conclude from (5.3.9) that $\frac{d\delta(\lambda)}{d\lambda} \geq 0$ and therefore

$$\delta(0) = f(X) \leq \delta(1) = f(Y),$$

from which monotonicity follows.

To prove that f is submodular, we pick $X \subset Y \subset \mathcal{S}$, $s \in \mathcal{S} \setminus Y$ and now define instead

$$\begin{aligned} \delta(\lambda) &:= g \left(Q_0 + Q_s + \sum_{k \in X} Q_k + \lambda \sum_{k \in Y \setminus X} Q_k \right) \\ &\quad - g \left(Q_0 + \sum_{k \in X} Q_k + \lambda \sum_{k \in Y \setminus X} Q_k \right), \end{aligned}$$

$\forall \lambda \in [0, 1]$. We now have

$$\begin{aligned} \delta(0) &= g \left(Q_0 + Q_s + \sum_{k \in X} Q_k \right) - g \left(Q_0 + \sum_{k \in X} Q_k \right) \\ &= f(X \cup \{s\}) - f(X), \end{aligned}$$

and

$$\begin{aligned}\delta(1) &= g\left(Q_0 + Q_s + \sum_{k \in X} Q_k + \sum_{k \in Y \setminus X} Q_k\right) \\ &\quad - g\left(Q_0 + \sum_{k \in X} Q_k + \sum_{k \in Y \setminus X} Q_k\right) \\ &= f(Y \cup \{s\}) - f(Y)\end{aligned}$$

and

$$\begin{aligned}\frac{d\delta(\lambda)}{d\lambda} &= -\text{trace}\left[\left(D_g\left(Q_0 + \sum_{k \in X} Q_k + \lambda \sum_{k \in Y \setminus X} Q_k\right)\right.\right. \\ &\quad \left.\left.- D_g\left(Q_0 + Q_s + \sum_{k \in X} Q_k + \lambda \sum_{k \in Y \setminus X} Q_k\right)\right)\right]^\top \\ &\quad \times \left[\sum_{k \in Y \setminus X} Q_k\right].\end{aligned}$$

Since $Q_0 \succ 0$ and $Q_s \succeq 0$, we conclude from (5.3.9) that

$$\begin{aligned}D_g\left(Q_0 + \sum_{k \in X} Q_k + \lambda \sum_{k \in Y \setminus X} Q_k\right) \\ - D_g\left(Q_0 + Q_s + \sum_{k \in X} Q_k + \lambda \sum_{k \in Y \setminus X} Q_k\right) \succeq 0\end{aligned}$$

and therefore

$$\frac{d\delta(\lambda)}{d\lambda} \leq 0,$$

because the trace of the product of two positive semidefinite matrices is non-negative.

This shows that

$$\delta(1) = f(Y \cup \{s\}) - f(Y) \leq f(X \cup \{s\}) - f(X) = \delta(0),$$

from which submodularity follows. \square

Proof of Theorem 5.1. The function in (5.3.2) is normalized since for $F_i = \emptyset$, the two terms cancel and therefore $f(\emptyset) = 0$. We prove that this function is monotone and submodular by applying Lemma 5.1 to the function

$$g(A) := \log \det(A) - \log \det(Q_0), \forall A \in \mathbb{R}^{n \times n},$$

for which (5.3.6) precisely matches (5.3.2). For this function g , we have that for any symmetric positive definite matrix $A \in \mathbb{R}^{n \times n}$

$$D_g(A) = A^{-T} > 0,$$

and therefore, for every pair of symmetric positive definite matrices $A, B \in \mathbb{R}^{n \times n}$, we have that

$$\begin{aligned} A \succeq B &\implies B^{-1/2} A B^{-1/2} \succeq I \\ &\implies \lambda_i [B^{-1/2} A B^{-1/2}] \geq 1 \\ &\implies \lambda_i [B^{1/2} A B^{1/2}] \leq 1 \\ &\implies B^{1/2} A B^{1/2} \preceq I \\ &\implies A^{-1} \preceq B^{-1}. \end{aligned}$$

This shows that we can indeed apply Lemma 5.1 and the result follows.

□

5.4 Motion Models

We consider a scenario in which the n mobile agents carry a suite of onboard sensors to estimate the trajectory of a vehicle and denote by $q(t)$ the vehicle's position at time t , expressed in an inertial coordinate system. We consider a constant curvature motion model for $q(t)$. Assuming that the vehicle's linear and angular velocities have constant coordinates v^b and ω^b , respectively, when expressed in the body frame, we have

$$\dot{q} = Rv^b,$$

and

$$\dot{R} = RJ(\omega^b),$$

with $R \in \text{SO}(3)$. For this model, the coordinates of the linear and angular accelerations expressed in the inertial frame satisfy the equations

$$\dot{v}^i = \dot{R}v^b = RJ(\omega^b)v^b = J(R\omega^b)Rv^b = J(\omega^i)v^i,$$

and

$$\dot{\omega}^i = \dot{R}\omega^b = RJ(\omega^b)\omega^b = 0,$$

which leads to the motion model

$$\dot{q} = v^i,$$

and

$$\dot{v}^i = J(\omega^i)v^i,$$

where we can view the angular velocity ω^i as an (unknown) constant parameter. This differential equation can be integrated exactly on a time interval $t \in [t_{\ell-1}, t_\ell]$ since

$$\begin{aligned} v^i(t) &= e^{J(\omega^i)(t-t_{\ell-1})} v^i(t_{\ell-1}) \\ &= \left(I + \sin(\rho(t-t_{\ell-1})) J(\bar{\omega}) \right. \\ &\quad \left. + (1 - \cos(\rho(t-t_{\ell-1}))) J(\bar{\omega})^2 \right) v^i(t_{\ell-1}), \end{aligned}$$

where

$$\bar{\omega} := \frac{\omega^i}{\|\omega^i\|},$$

and

$$\rho := \|\omega^i\|.$$

Since the exact formula for $q(t)$ is complex, we use its 2nd order Taylor series approximation for t close to $t_{\ell-1}$, which leads to

$$v^i(t) = v^i(t_{\ell-1}) + (t - t_{\ell-1}) J(\omega^i) v^i(t_{\ell-1}),$$

and

$$\begin{aligned} q(t) &= q(t_{\ell-1}) + (t - t_{\ell-1}) v^i(t_{\ell-1}) \\ &\quad + \frac{(t - t_{\ell-1})^2}{2} J(\omega^i) v^i(t_{\ell-1}). \end{aligned} \tag{5.4.1}$$

This motion model can be summarized as

$$q(t) = \theta_1 + (t - t_{\ell-1}) \theta_2 + \frac{(t - t_{\ell-1})^2}{2} \theta_3, \tag{5.4.2}$$

where $\theta_1, \theta_2, \theta_3 \in \mathbb{R}^2$ can be viewed as three parameters that need to be estimated. These parameters can be viewed as the target's position, linear velocity, and curvature on the interval $t \in [t_{\ell-1}, t_\ell]$. For targets moving in along a straight line, this model simplifies to the case $\theta_3 = 0$ and for stationary targets $\theta_2 = \theta_3 = 0$.

5.5 Measurement Models

We denote by $p(t) \in \mathbb{R}^3$ the known position of a sensing agent that collects measurements to estimate the vehicle's trajectory. In this section, we specifically consider on-board RF sensors that measure (i) the times of arrival of radio packets emitted by the vehicle; (ii) the Doppler frequency shift in their carrier frequency arising from the relative motion between the vehicle and receiver; and (iii) image coordinates of distinguishable features of the vehicle, collected by on-board visible or IR cameras. With regard to the RF measurements, we do not assume the vehicle's transmissions are synchronized with the receiver's clock nor precise knowledge of the vehicle's carrier frequency, which essentially means that our measurements should be viewed as time difference of arrival (TDoA) and frequency difference of arrival (FDoA). Because of the TDoA and FDoA ambiguity, in addition to the motion model parameters $\theta_1, \theta_2, \theta_3$ in (5.4.2), we also need to estimate sensor-specific parameters that account for the lack of synchronization and knowledge of the carrier frequency.

The remainder of this section, discusses the different sensor measurement models and implicitly assume that the different sensors produce conditionally independent measurements, given all the parameters that need to be estimated. We also assume here that measurements y_k occur at times $\tau_k \in [t_{\ell-1}, t_\ell]$ and have multi-variable normal (conditional) distributions with mean $\mu_k(\theta)$ that depends on the vector θ of unknown parameters and covariance Σ_k that, for simplicity, does not depend on θ . In this case, the matrix

Q_k in (5.2.3) is given by

$$Q_k = \frac{\partial \mu_k(\theta)}{\partial \theta} \Sigma_k^{-1} \frac{\partial \mu_k(\theta)}{\partial \theta}^\top,$$

[95], where $\frac{\partial \mu_k(\theta)}{\partial \theta}$ denotes the Jacobian matrix of μ_k . Because of the nonlinearity of the map $\theta \mapsto \mu_k(\theta)$, the a-posteriori distribution of θ given such measurements will typically be non-Gaussian and often multi-modal.

5.5.1 Time-of-Arrival Measurements

The vehicle's radio transmitter sends symbols at times $\tau_k \in [t_{\ell-1}, t_\ell]$

$$\tau_k = kT + T_0,$$

with $k \in \{0, 1, \dots, K-1\}$ where T is only approximately known and T_0 is unknown to the receiver. Note that T_0 need not be the same as the initial time of the estimation interval, $t_{\ell-1}$. The receiver records noisy observations of the times-of-arrival of the symbols, denoted by \mathcal{T} , which is given by

$$\mathcal{T}(\tau_k) = \tau_k + \frac{\rho(\tau_k)}{c},$$

where c denotes the speed of light and

$$\rho(t) := \|q(t) - p(t)\|. \quad (5.5.1)$$

Therefore, the times-of-arrival scaled by the speed of light are given by

$$c\mathcal{T}(\tau_k) = \rho(\tau_k) + \theta_T k + \theta_{T_0}, \quad (5.5.2)$$

where

$$\theta_T := cT,$$

and

$$\theta_{T_0} := cT_0.$$

This model assumes that the relative motion between transmitter and receiver is sufficiently slow so that the receiver's position at the time the symbol is received is essentially the same as when it was transmitted. We regard the noisy measurements of the times-of-arrival as Gaussian random variables with means given by the actual times-of-arrival in (5.5.2) and variance $\sigma_{c\mathcal{T}}^2$ independent of the unknown parameters, which means that the likelihood of a measurement y_k of (5.5.2) is given by

$$P(y_k; \sigma_{c\mathcal{T}}, \theta_T, \theta_{T_0}) = \frac{1}{\sqrt{8\pi}\sigma_{c\mathcal{T}}} e^{-\frac{(\rho(\tau_k) + \theta_T k + \theta_{T_0} - y_k)^2}{2\sigma_{c\mathcal{T}}^2}}.$$

The parameters θ_T and θ_{T_0} are typically not known a priori and must be estimated jointly along with θ_1 , θ_2 , and θ_3 . For the motion model in (5.4.2), the gradient with respect to the motion-specific parameters is given by

$$\frac{\partial c\mathcal{T}(\tau_k)}{\partial \theta_1} = \frac{(q(\tau_k) - p(\tau_k))}{\rho(\tau_k)},$$

and

$$\frac{\partial c\mathcal{T}(\tau_k)}{\partial \theta_2} = \frac{(q(\tau_k) - p(\tau_k))}{\rho(\tau_k)} (\tau_k - t_{\ell-1}),$$

and

$$\frac{\partial c\mathcal{T}(\tau_k)}{\partial \theta_3} = \frac{(q(\tau_k) - p(\tau_k)) (\tau_k - t_{\ell-1})^2}{\rho(\tau_k) 2}.$$

The gradient with respect to the sensor specific parameters θ_T , and θ_{T_0} , is given as

$$\frac{\partial c\mathcal{T}(\tau_k)}{\partial \theta_T} = k,$$

and

$$\frac{\partial c\mathcal{T}(\tau_k)}{\partial \theta_{T_0}} = 1.$$

5.5.2 Doppler Measurements

The vehicle's radio decoder detects the frequency shift of the received carrier, which results from both a mismatch between transmitter and receiver center frequencies as well as from the relative motion between transmitter and receiver. Specifically, the frequency shift associated with the k th symbol is given by for $\tau_k \in [t_{\ell-1}, t_\ell]$

$$\mathcal{F}(\tau_k) = \Delta f(\tau_k) - \frac{\dot{\rho}(\tau_k)}{\lambda}, \quad (5.5.3)$$

where $\lambda = \frac{c}{f_c}$, f_c is the carrier frequency of the transmitter, Δf is the difference between the carrier frequencies of the transmitter and receiver, and ρ is defined in (5.5.1), leading to

$$\dot{\rho}(t) = \frac{(q(t) - p(t))^\top (\dot{q}(t) - \dot{p}(t))}{\|q(t) - p(t)\|}.$$

Therefore, the frequency shifts (5.5.3) scaled by the wave length are given by

$$\lambda\mathcal{F}(\tau_k) = \theta_\lambda - \dot{\rho}(\tau_k), \quad (5.5.4)$$

where

$$\theta_\lambda := \lambda\Delta f.$$

We regard the noisy measurements of the frequency shifts as Gaussian random variables with means given by the actual frequency shifts in (5.5.4) and variance $\sigma_{\lambda\mathcal{F}}^2$ independent of the unknown parameters, from which follows that the likelihood of a measurement y_k of (5.5.4) is given by

$$P(y_k; \sigma_{\lambda\mathcal{F}}, \theta_\lambda) = \frac{1}{\sqrt{2\pi}\sigma_{\lambda\mathcal{F}}} e^{-\frac{(\theta_\lambda - \dot{p}(\tau_k) - y_k)^2}{2\sigma_{\lambda\mathcal{F}}^2}}.$$

Here, the parameter θ_λ is typically not know and also needs to be estimated. For the motion model in (5.4.2), the gradient with respect to the motion parameters is given by

$$\begin{aligned} \frac{\partial \lambda\mathcal{F}(\tau_k)}{\partial \theta_1} &= \frac{(\dot{q}(\tau_k) - \dot{p}(\tau_k))^\top}{\|q(\tau_k) - p(\tau_k)\|} \\ &\quad \times \left(I - \frac{(q(\tau_k) - p(\tau_k))(q(\tau_k) - p(\tau_k))^\top}{\|q(\tau_k) - p(\tau_k)\|^2} \right), \end{aligned}$$

and

$$\begin{aligned} \frac{\partial \lambda\mathcal{F}(\tau_k)}{\partial \theta_2} &= \frac{(\dot{q}(\tau_k) - \dot{p}(\tau_k))^\top}{\|q(\tau_k) - p(\tau_k)\|} \\ &\quad \times \left(I - \frac{(q(\tau_k) - p(\tau_k))(q(\tau_k) - p(\tau_k))^\top}{\|q(\tau_k) - p(\tau_k)\|^2} \right) \\ &\quad \times (\tau_k - t_{k-1}) \\ &\quad + \frac{(q(\tau_k) - p(\tau_k))^\top}{\|q(\tau_k) - p(\tau_k)\|}, \end{aligned}$$

and

$$\begin{aligned} \frac{\partial \lambda \mathcal{F}(\tau_k)}{\partial \theta_3} &= \frac{(\dot{q}(\tau_k) - \dot{p}(\tau_k))^\top}{\|q(\tau_k) - p(\tau_k)\|} \\ &\times \left(I - \frac{(q(\tau_k) - p(\tau_k))(q(\tau_k) - p(\tau_k))^\top}{\|q(\tau_k) - p(\tau_k)\|^2} \right) \\ &\times \frac{(\tau_k - t_{k-1})^2}{2} \\ &+ \frac{(q(\tau_k) - p(\tau_k))^\top}{\|q(\tau_k) - p(\tau_k)\|} (\tau_k - t_{k-1}). \end{aligned}$$

The gradient with respect to θ_λ is given as

$$\frac{\partial \lambda \mathcal{F}(t_k)}{\partial \theta_\lambda} = 1.$$

5.5.3 Camera Measurements

The sensing agent has an on-board camera and associated image-processing algorithms that determine the target's image coordinates. Assuming a projective camera model with optical center at the agent's position $p(\tau_k)$ and a (known) camera orientation $R_c(\tau_k)$, the target's image coordinates at time $\tau_k \in [t_{\ell-1}, t_\ell]$ are given as [96, Chapter. 5, p. 141]

$$\mathcal{I}(\tau_k) = \frac{M(\tau_k)(q(\tau_k) - p(\tau_k))}{m(\tau_k)(q(\tau_k) - p(\tau_k))}, \quad (5.5.5)$$

where

$$M(t) := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} AR_c(t),$$

and

$$m(t) := \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} AR_c(t),$$

and A denotes the camera's intrinsic parameters matrix, which is defined as

$$A = \begin{bmatrix} f_x & s_{xy} & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix},$$

where f_u, f_v are the camera focal lengths in the x , and y directions, respectively, in the image plane; (o_x, o_y) is the focal center of the image plane and s_{xy} is the skew parameter. We regard the target's image coordinates produced by the image processing algorithms as Gaussian random vectors in \mathbb{R}^2 with means given by (5.5.5) and covariance matrix $\Sigma_{\mathcal{I}}$ independent of the unknown parameters, which means that the likelihood of a measurement y_k of (5.5.5) is given by

$$P(y_k; \Sigma_{\mathcal{I}}) = \frac{1}{2\pi\sqrt{\det \Sigma_{\mathcal{I}}}} e^{-\frac{1}{2}(\mathcal{I}(\tau_k) - y_k)^\top \Sigma_{\mathcal{I}} (\mathcal{I}(\tau_k) - y_k)}.$$

For the motion model in (5.4.2), the gradient with respect to the motion parameters is given by

$$\frac{\partial \mathcal{I}(\tau_k)}{\partial \theta_1} = \frac{M(\tau_k)}{d(\tau_k)} - \frac{U(\tau_k)}{d(\tau_k)^2} m(\tau_k),$$

and

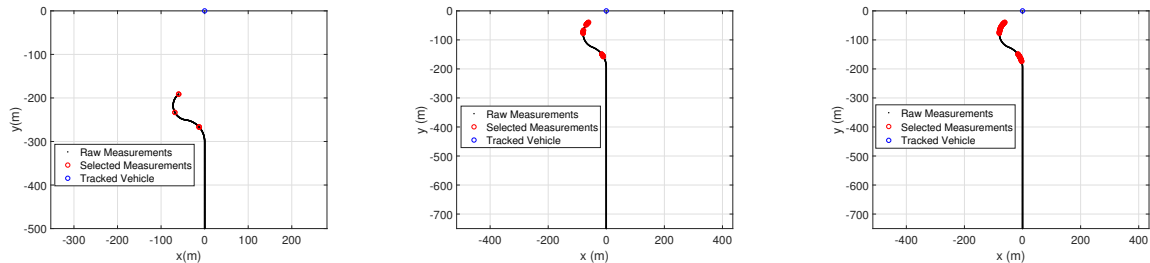
$$\frac{\partial \mathcal{I}(\tau_k)}{\partial \theta_2} = (\tau_k - t_{k-1}) \left(\frac{M(\tau_k)}{d(\tau_k)} - \frac{U(\tau_k)}{d(\tau_k)^2} m(\tau_k) \right),$$

and

$$\frac{\partial \mathcal{I}(\tau_k)}{\partial \theta_3} = \frac{(\tau_k - t_{k-1})^2}{2} \left(\frac{M(\tau_k)}{d(\tau_k)} - \frac{U(\tau_k)}{d(\tau_k)^2} m(\tau_k) \right),$$

where

$$U(t) := M(t)(q(t) - p(t)),$$



(a) 10 measurements selected. (b) 50 measurements selected. (c) 100 measurements selected.

Figure 5.6.1: Example 1, where 1000 total measurements are collected uniformly along the path shown in black. Small subsets of measurements are selected using the proposed methodology and are shown in red. The ground-truth location of the object to be tracked is marked in blue.

and

$$d(t) := m(t)(q(t) - p(t)).$$

5.6 Results

This section contains several examples based on synthetic data that illustrate the benefits of FIM-based measurement selection.

5.6.1 Example 1: A Single Camera

Our first example is that of an agent with a single imaging sensor, measuring the position of a stationary vehicle in the image plane as described in Sec. 5.5.3. The camera has a focal length of 50 pixels and image-plane noise with a standard deviation of 0.8 pixels. An agent follows the path shown in Fig. 5.6.1. This example highlights how the FIM can be used to assess how informative each of the vision measurements are. In this instance, when the path is facing straight at the object then little information is gained in the 'y' direction since any single image cannot measure depth. However, when multiple images are captured at different angles relative to the object, then depth can be estimated. The

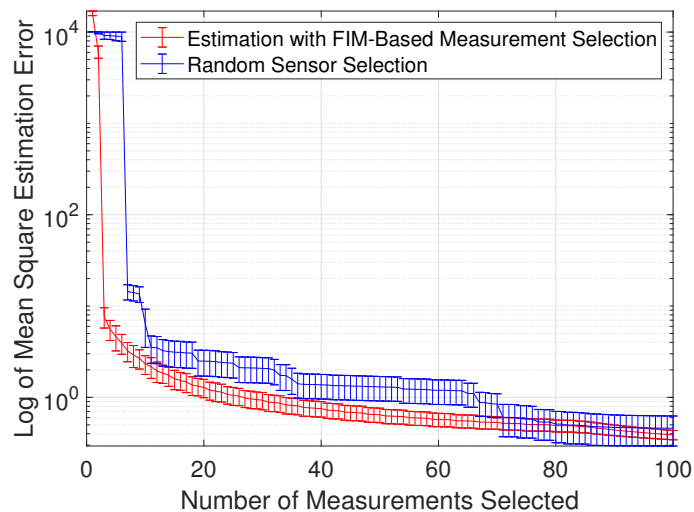
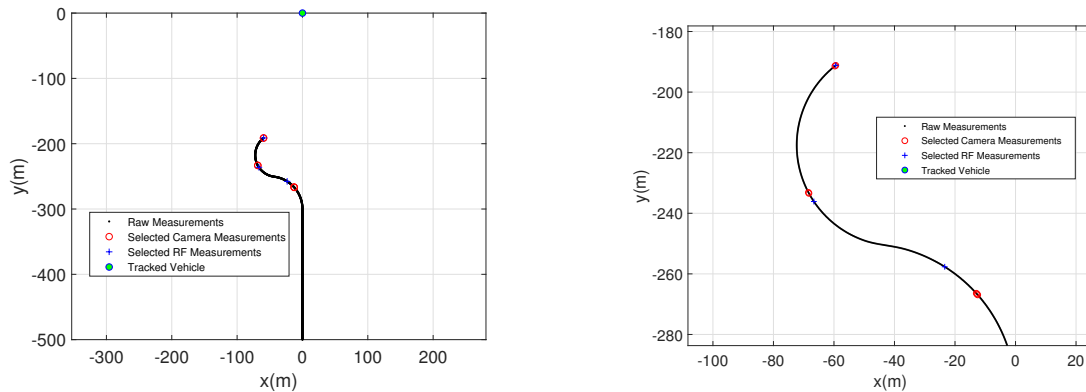


Figure 5.6.2: Averaged estimation error versus the number of measurements selected for Example 1. Red shows the average estimation error using measurement selected using the FIM-based method in Algorithm 2 and blue show the average estimation error using a random selection.

FIM quantifies this phenomenon and we can optimally pick the most informative subset of measurements. This is shown in Fig. 5.6.1 where 1000 measurements are uniformly collected along the agent’s path and a small subset of measurements are selected using Algorithm 2 to pick those measurements that minimally reduce the degradation of estimation performance as compared to the full set of collected measurements. The FIM-selected measurements provide a compromise between closeness to the target (which is maximal right at the end of the path) and largest diversity of viewing angle. We use the unscented transform [34, 97] to estimate the parameters and the resulting estimation error is shown in Figure 5.6.2, though other state-of-the-art estimation schemes designed specifically for tracking could also be utilized such as interacting multiple model methods [98] or changepoint filtering [99]. Figure 5.6.2 shows tremendous reduction in estimation error when using FIM-based selection instead of random selection for regimes with very few measurements selected.



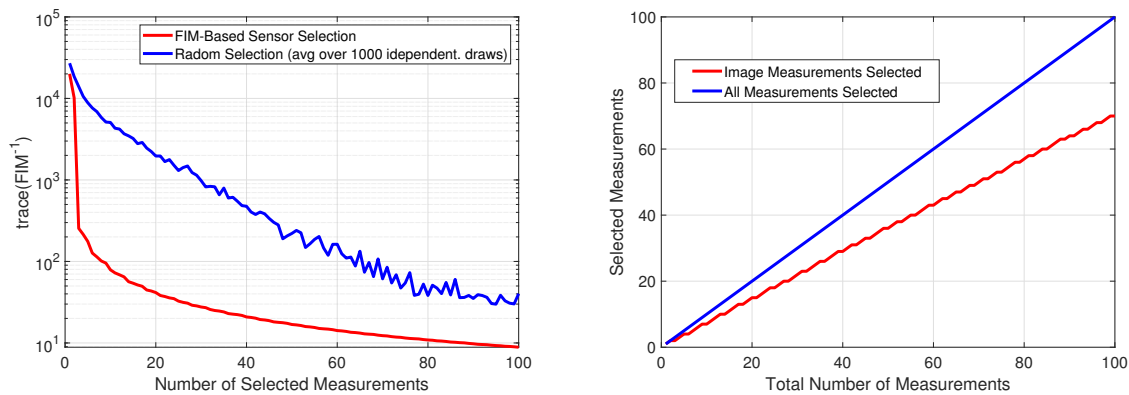
(a) 10 measurements selected from 1000 measurements collected each from a camera and RF sensor.

(b) Zoomed to show detail of the measurements selected during the final section of the trajectory.

Figure 5.6.3: Example 2, where 1000 total measurements are collected each from a camera and RF sensor, uniformly along the path shown in black. A subset of 10 measurements are selected between the sensors using the proposed methodology. Selected camera measurements are shown red and selected RF measurements are shown in blue. 7 vision measurements and 3 RF measurements were selected out of 1000 total available. The ground-truth location of the object to be tracked is marked in green.

5.6.2 Example 2: Two Heterogeneous Sensors

The next example is based on similar trajectories for the vehicle and mobile agent as in Example 1, but the latter has an additional RF sensor that measures the Doppler shift, as in Section 5.5.2, with a noise standard deviation of 33ppb in frequency. The agent uses the FIM-based criteria to select a mixture of measurements between the two sensors. The performance is compared to randomly selecting measurements and is shown in the Figure 5.6.4a, where we see a dramatic reduction in estimation error with just a small number of measurements selected. Figure 5.6.4b shows the ratio of image measurements to RF measurements selected. Where we can see that, for this geometry FIM-based selections roughly pick 2/3 of the measurements from the camera versus 1/3 from the radio receiver.



(a) Performance versus the number of total measurements selected from a pool of 1000 measurements collected each from a camera and RF sensor. (b) Ratio of RF and image measurements selected.

Figure 5.6.4: Example 2, where measurements are jointly selected from two heterogeneous sensors.

5.6.3 Example 3: Multiple Platforms

We expand on Example 2 by having two agents, each with a camera and RF sensor measuring frequency shifts. Each agent selects a mixture of camera and RF measurements based in Algorithm 2 and sends it to a centralized node for processing. Figure 5.6.5 shows which measurements are selected from each agent. The error covariance versus the number of measurements selected by each agent is shown in Figure 5.6.6. Compared to estimation with only a single agent, multiple independent agents performing FIM-based measurement selection performs better.

5.6.4 Cooperative Measurement Selection

The measurement selection algorithm provided by Algorithm 2 operates independently across agents and therefore does not require inter-agent communication. However, when communication between agents is available, there is opportunity for further performance improvement, even when only a small amount of information can be exchanged

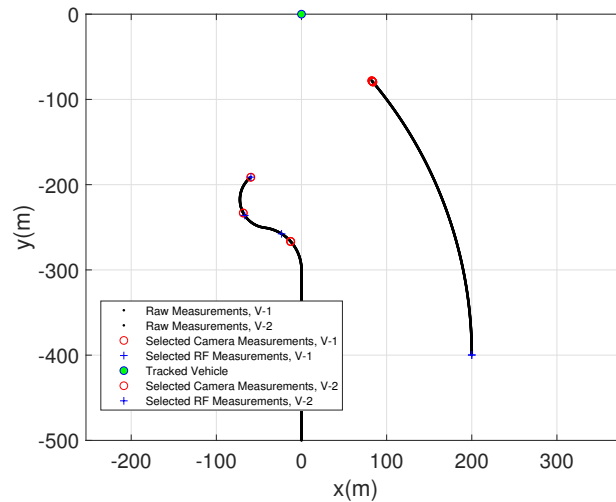


Figure 5.6.5: Example 3, where each of the 2 agents collects 1000 total measurements each from a camera and RF sensor, uniformly along the paths shown in black. A subset of 10 measurements are selected between the sensors using the proposed methodology from each agent. Selected camera measurements are shown red and selected RF measurements are shown in blue. The ground-truth location of the object to be tracked is marked in green.

between agents.

Consider Example 3 above, but after agent 1 chooses a set of measurements, the FIM matrix $FIM(F_i^*)$ is shared with agent 2 to use as a replacement of Q_0 in its local selection of measurements, F_2^* . This allows the consideration of agent 1's selection to aid in selecting a better subset from agent 2. The result is a substantial decrease in estimation error compared to the independent selection and is shown in Figure 5.6.6.

5.7 Conclusions and Future Work

We presented a FIM-based submodular criteria to select measurements for near-optimal estimation performance in a computationally efficient manor. We construct the FIM for several sensors that are commonly used in vehicle tracking problems. Future work includes establishing theoretical guarantees for cooperative sensor selection and val-

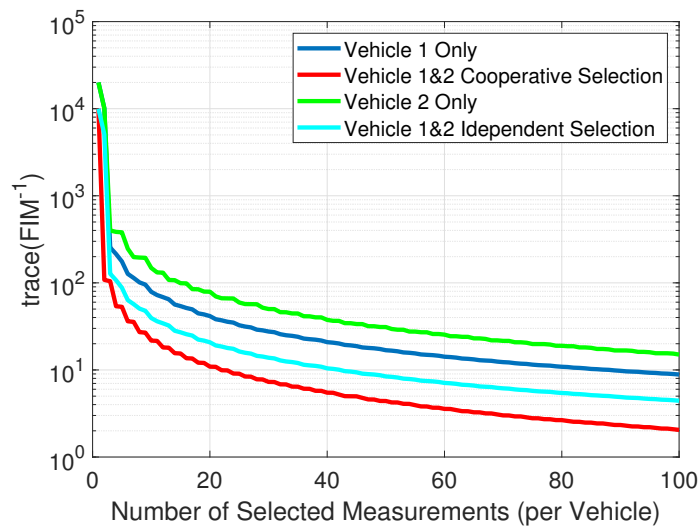


Figure 5.6.6: Estimation error performance using only the given number of measurements selected using the FIM-based approach. The blue and green show the performance for using only agent 1 or 2, respectively. The cyan curve is the performance when both are used at the centralized node to improve estimation after being selected independently with a FIM-based criteria. Red is the performance when cooperative FIM-based measurement selection is performed.

validation on experimental data. We have preliminary results showing that the cooperative algorithm outlined in Section 5.6.4 provides theoretical guarantees of performance when compared to the optimal centralized algorithm.

Chapter 6

Trajectories for the Optimal Collection of Information

6.1 Introduction

We present a method to optimize vehicle trajectories to gain maximal information for target tracking problems. The scenario currently being studied is an aircraft receiving passive information from sensors rigidly mounted to the airframe. These sensors include, but are not limited to, infrared or visible spectrum, as well as RF receivers that measure the frequency shifts from an external transmitter. The measurements are sampled in order to determine the location of a target vehicle. The placement of the sensors is determined by the path of the aircraft, influencing how much information is gained as well as the overall effectiveness of estimating where the target is located. By optimizing the trajectory, we can achieve maximum information gain, and hence the greatest accuracy in localizing the target.

This problem is a generalization of what appeared in [100], where the path of the vehicle was fixed and a subset of measurements were selected only from along this path.

In this context we optimize a metric of the cumulative Fisher Information Matrix (FIM) of the aircraft path, which is motivated by its connection to the (Bayesian) Cramér-Rao lower bound [92]. The LOGDET metric is chosen as this gives a D-optimal estimate, essentially corresponding to minimizing the volume of the error ellipsoid, and additionally provides favorable numeric properties. It is worth noting that while the focus of this paper is the LOGDET metric, other metrics may be considered, provided the metric meets certain conditions that are outlined in what follows in the paper. Of particular interest would be the trace of the inverse metric, as that gives the A-optimal estimate, effectively minimizing the mean-square estimate error. Analysis of the trace of the inverse metric is outside the scope of this paper and will be investigated in future work.

We formulate the problem in such a way that the optimal value function satisfies a Hamilton-Jacobi (HJ) partial differential equation (PDE), from which the optimal trajectories immediately follow. Naively, a solution of the corresponding HJ PDE using a grid-based method would have many advantages since they handle the non-linear and non-convex problems that arises in FIM-based optimization. However, the sensor estimation problem induces a state space dimension that renders typical grid-based methods [101] for PDE solutions intractable due to the exponential dimensional scaling of such methods. Recognition of this problem is not new, and the phrase “curse of dimensionality” was coined decades ago by Richard Bellman [102]. This creates a large gap between the rigorous theory of HJ equations and practical implementation on many problems of interest, especially vehicle planning and coordination problems.

New research has emerged in an attempt to bridge this technological gap, including trajectory optimization approaches [10, 11, 73], machine learning techniques [103, 104, 105], and sub-problem decomposition [106, 107]. The structure of the sensor placement problem lends itself well to the later strategy. Unique in this context, though, is that we do not need to abandon spatial grids entirely, instead forming a hybrid approach. This

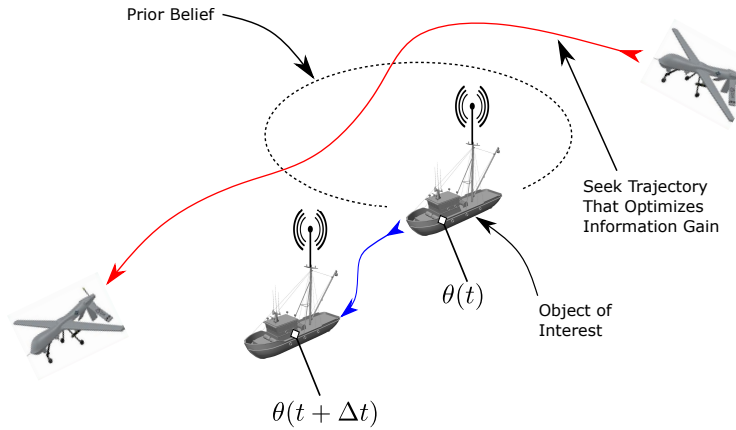


Figure 6.1.1: An illustration of the target tracking problem. An aircraft collects measurement for sensors as it flies along a path, attempting to estimate the location of the ship, denoted here as θ . Modifying the path of the vehicle can greatly improve the estimation performance.

leverages the strength of grid-based methods in dealing with the non-convexities that commonly arise when using the FIM matrix, but restricts their applications to a small subspace of the problem.

In what follows we formally introduce the sensor estimation problem and form its corresponding HJ PDE. We then proceed to show a new hybrid method of lines (MOL) approach that involves decomposing the state space. and conclude with simulated results of the optimal trajectories that result from heterogeneous sensors tracking the location of a mobile target. Section 2 shows how the information collecting problem gives rise to nonlinear dynamics with a cascade structure, that the input only directly affects one first subcomponent of the state, whereas the optimization criteria only depends on a second subcomponent. Section 3 addresses the optimal control of this type of systems using the HJ PDE and the classical MOL. Section 4, develops the theory needed for the new hybrid method of lines, which is applicable to systems in a cascade form. This type of systems arises naturally in formation collecting, but the hybrid methods of lines can be applied to the optimal of more general cascade systems. Section 5 specializes the hybrid MOL to the information collection. Section 6 includes simulation results for a particular

vehicle model and sensor type.

6.2 The Vehicle Sensing Problem

We choose as our vehicle a Dubin's car [44] and denote by $(X, Y, \psi) := x \in \mathcal{X} := \mathbb{R}^2 \times \text{SO}(2)$ the vehicle state where X and Y are the rectangular positional coordinates of the vehicle center and ψ is the heading angle. The dynamics are defined by

$$\frac{d}{ds}x(s) = f(x(s)) + Bu(s), \text{ a.e. } s \in [0, t] \quad (6.2.1)$$

where

$$f(x) = \begin{bmatrix} v \cos \psi \\ v \sin \psi \\ 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad (6.2.2)$$

where $u(s) \in U := [-\omega_{\max}, \omega_{\max}]$ is the allowable control set of turn rates and v is the fixed forward speed of the vehicle. The admissible control set is defined as

$$U[0, t] := \{u(\cdot) : [0, t] \rightarrow U \mid u(\cdot) \text{ is measurable}\}. \quad (6.2.3)$$

Our method applied to vehicles that can be expressed in the general form (6.2.1), which includes the Dubins vehicle in (6.2.2). The Dubins vehicle with bounded turning rate is particularly interesting because it is a low-dimensional model that generates trajectories that are easy to track by an aircraft flying at constant speed and altitude.

The vehicle defined above has a group of rigidly attached sensors collecting measurements. The measurements, denoted as y , are sampled in order to determine an unknown random variable, θ . The measurements are assumed to be random variables dependent

on θ with density function

$$y \sim \rho(y|\theta).$$

Assuming that all measurements y are conditionally independent given θ , the cumulative Bayesian Fisher Information Matrix (FIM) associated with the estimation of θ is of the form

$$\text{FIM}(t, x, u(\cdot)) := Q_0 + \int_0^t Q(\gamma(s; x, u(\cdot))) ds,$$

where

$$Q(x) := \mathbb{E}_\theta [Q(x; \theta)], \quad (6.2.4)$$

with

$$Q(x; \theta) := \mathbb{E}_y \left[\left(\frac{\partial \log \rho(y|\theta, x)}{\partial \theta} \right)^\top \left(\frac{\partial \log \rho(y|\theta, x)}{\partial \theta} \right) \right], \quad (6.2.5)$$

and

$$Q_0 := \mathbb{E}_\theta \left[\left(\frac{\partial \log \rho(\theta)}{\partial \theta} \right)^\top \left(\frac{\partial \log \rho(\theta)}{\partial \theta} \right) \right],$$

where $\rho(\theta)$ is the a-priori probability density function for θ . The formula above assumes a scenario where the measurement, $y(t)$, is collected by one sensor or by multiple independent sensors that generate at the same (constant) sampling rate. When multiple independent sensors collect measurements at constant but different sampling rates, the FIM matrix can be factored for each sensor i :

$$Q(t, x, u(\cdot)) = \sum_i F^i Q^i(\gamma(s; x, u(\cdot))),$$

where F^i is the sampling rate of the i -th sensor. The above matrices are given from [108], where the expectation over y in (6.2.5) is given in closed form for some distributions, see for example [100, Sec. 5]. While the outer expectation over θ in (6.2.4) is rarely known

in closed form, many approximation schemes can be employed, for example Monte Carlo sampling or Taylor series expansion.

The placement of the sensors is determined by the path of the aircraft, influencing how much information is gained as well the overall effectiveness of estimating θ . Therefore we optimize the trajectory to achieve maximum information gain, and hence the greatest performance in estimating θ from the measurements y . For a given initial state $x \in \mathcal{X}$ and terminal time $t \in [0, \infty)$, we define the following cost functional:

$$J(t, x, u(\cdot)) := G(\text{CFIM}(t, x, u(\cdot))) + \log \det(Q_0), \quad (6.2.6)$$

where

$$G(x, Z) = G(Z) := -\log \det(Z),$$

We denote by $V(t, x)$ the value function defined as

$$V(t, x) = \inf_{u(\cdot) \in U[0, t]} J(t, x, u(\cdot)), \quad (6.2.7)$$

which can be interpreted as the maximal information gain for a family of trajectory optimization problems parameterized by initial state $x \in \mathcal{X}$ and terminal time $t \in [0, \infty)$.

The cost functional in (6.2.7) is not in a standard form, so we convert the problem into a common standard, the so-called Mayer form. To do this we augment the state vector with $z := \text{vec}(Z)$, where the matrix $Z \in \mathcal{Z} := \text{dom}(G)$. Our new state becomes

$$\chi := (x, z)^\top,$$

with augmented dynamics

$$\frac{d}{ds}\chi(s) = \hat{f}(\chi(s), u(s)) = \begin{bmatrix} f(x(s)) \\ \ell(x(s)) \end{bmatrix} + \begin{bmatrix} B \\ \mathbf{0} \end{bmatrix} u(s), \quad (6.2.8)$$

with

$$\ell(x(s)) := \text{vec}(Q(x(s))),$$

where vec is the vectorize operator that reshapes a matrix into a column vector and $\mathbf{0}$ is a vector of zeros of the same number of elements as the augmented variable z . If we fix the z initial condition such that

$$z = \text{vec}(Q_0), \quad (6.2.9)$$

then the cost functional (6.2.6) can equivalently written as

$$J(t, x, u(\cdot)) = J(t, \chi, u(\cdot)) = G(\text{vec}^{-1}(z)), \quad (6.2.10)$$

where we denote by $Z = \text{vec}^{-1}(z)$ the inverse operator such that

$$\text{vec}(\text{vec}^{-1}(z)) = z.$$

Hereafter we will denote by \tilde{G} as the function G with the input reshaped as a function of z with

$$\tilde{G}(z) := G(\text{vec}^{-1}(z)). \quad (6.2.11)$$

Likewise the value function is equivalently written as

$$V(t, \chi) = \inf_{u(\cdot) \in U[0,t]} J(t, \chi, u(\cdot)). \quad (6.2.12)$$

6.3 Decomposition of Coupled Systems

The approach we will develop to solve (6.2.12) is applicable to a more general class of cascade systems that we introduce in this section, and for which we discuss the use of HJ methods for optimal control. Denote by $\chi := (x, z)^\top$ where $x \in \mathcal{X} = \mathbb{R}^n$ and $z \in \mathcal{Z} = \mathbb{R}^m$. The state has coupled dynamics as follows:

$$\begin{cases} \dot{x}(s) = f(x(s)) + g(x(s))u(s) & \text{a.e } s \in [0, t] \\ \dot{z}(s) = \ell(x(s)), \end{cases} \quad (6.3.1)$$

with $u \in U$, where U is a closed convex set. We denote by $[0, t] \ni s \mapsto \gamma(s; x_0, u(\cdot)) \in \mathbb{R}^n$ the x state trajectory that evolves in time according to (6.2.1) starting from initial state x_0 at $t = 0$. The trajectory γ is a solution of (6.2.1) in that it satisfies (6.2.1) almost everywhere:

$$\begin{cases} \dot{\gamma}(s; x_0, u(\cdot)) = f(\gamma(s; x_0, u(\cdot))) + g(\gamma(s; x_0, u(\cdot)))u, \\ \gamma(0; x_0, u(\cdot)) = x_0. \end{cases} \quad (6.3.2)$$

Likewise, we denote by $[0, t] \ni s \mapsto \xi(s; \chi_0, u(\cdot))$ the trajectory of the z variable and it satisfies the following almost everywhere:

$$\begin{cases} \frac{d}{ds}\xi(s; \chi_0, u(\cdot)) = \ell(\gamma(s; x_0, u(\cdot))), \\ \xi(0; \chi_0, u(\cdot)) = z_0. \end{cases} \quad (6.3.3)$$

Note that the trajectory can be found directly from the expression:

$$\xi(s; \chi_0, u(\cdot)) := z_0 + \int_0^s \ell(\gamma(\tau; x_0, u(\cdot))) d\tau. \quad (6.3.4)$$

Denote $G : \mathbb{R}^m \rightarrow \mathbb{R}$ as the terminal cost function such that the mapping

$$\mathcal{Z} \ni z \mapsto G(z) \in \mathbb{R},$$

We define the cost functional

$$J(t, \chi, u(\cdot)) := G(\xi(t; \chi, u(\cdot))),$$

and the associated value function as

$$V(t, \chi) := \inf_{u(\cdot) \in U[0, t]} J(t, \chi, u(\cdot)),$$

where $U[0, t]$ is defined as in (6.2.3).

We denote by

$$\hat{f}(\chi, u) := \begin{bmatrix} f(x) + g(x)u \\ \ell(x) \end{bmatrix},$$

the joint vector field in (6.3.1). We assume that \hat{f} , U , and G satisfy the following regularity assumptions:

(F1) (U, d) is a separable metric space.

(F2) The maps $\hat{f} : \mathcal{X} \times U \rightarrow \mathbb{R}^{n+m}$ and $G : \mathcal{Z} \rightarrow \mathbb{R}$ are measurable, and there exists a constant $L > 0$ and a modulus of continuity $\omega : [0, \infty) \rightarrow [0, \infty)$ such that for $\varphi(\chi, u) = \hat{f}(\chi, u), G(z)$, we have for all $\chi, \chi' \in \mathcal{X} \times \mathcal{Z}$, and $u, u' \in U$

$$|\varphi(\chi, u) - \varphi(\chi', u')| \leq L \|\chi - \chi'\| + \omega(d(u, u')),$$

and

$$|\varphi(\mathbf{0}, u)| \leq L.$$

(F3) The maps \hat{f} , and G are C^1 in χ , and there exists a modulus of continuity $\omega : [0, \infty) \rightarrow [0, \infty)$ such that for $\varphi(\chi, u) = \hat{f}(\chi, u), G(z)$, we have for all $\chi, \chi' \in \mathcal{X} \times \mathcal{Z}$, and $u, u' \in U$

$$|\varphi_\chi(\chi, u) - \varphi_\chi(\chi', u')| \leq \omega(\|\chi - \chi'\| + d(u, u')).$$

6.3.1 Hamilton–Jacobi Formulation

Under a set of mild Lipschitz continuity assumptions, there exists a unique value function (6.2.12) that satisfies the following Hamilton–Jacobi (HJ) equation [18] with $V(t, \chi)$ being the viscosity solution of the partial differential equation (PDE) for $s \in [0, t]$

$$V_s(s, \chi) + \mathcal{H}(\chi, V_\chi(s, \chi)) = 0, \quad (6.3.5)$$

$$V(0, \chi) = G(z),$$

where $\sigma := (p, \lambda)^\top$ and

$$\mathcal{H}(\chi, \sigma) := \min_{u \in U} H(\chi, u, \sigma), \quad (6.3.6)$$

with the Hamiltonian, H , defined by

$$\begin{aligned} H(\chi, u, \sigma) &= \left\langle \begin{bmatrix} f(x) + g(x)u \\ \ell(x) \end{bmatrix}, \begin{bmatrix} p \\ \lambda \end{bmatrix} \right\rangle \\ &= \langle f(x), p \rangle + \langle g(x)u, p \rangle + \langle \ell(x), \lambda \rangle. \end{aligned}$$

In the case where the set U is bounded by a norm, i.e.

$$U = \{u \in \mathbb{R}^{n_u} \mid \|u\| \leq c\}, \quad (6.3.7)$$

for some c , then (6.3.6) is given in closed form by

$$\mathcal{H}(\chi, \rho) = \langle f(x), p \rangle + \left\| g(x)^\top p \right\|_* + \langle \ell(x), \lambda \rangle, \quad (6.3.8)$$

where $\|(\cdot)\|_*$ is the dual norm to $\|(\cdot)\|$ in (6.3.7). We denote by π the control that optimizes the Hamiltonian and is given by

$$\pi(s, \chi) := \arg \min_{u \in U} H(\chi, u, V_\chi(s, \chi)).$$

We note here that under mild assumptions, the viscosity solution of (6.3.5) is Lipschitz continuous in both s and χ [109, Theorem 2.5, p. 165]. This implies by Rademacher's theorem [110, Theorem 3.1.6, p. 216] the value function is differentiable almost everywhere. For what follows, we assume that the value function has continuous first and second derivatives. The points where this fails to be true only exists on a set of measure zero, and any practical implementation of the method presented will only evaluate points where the first and second derivatives exist. A characterization of the differentiability of the value function is outside the scope of this paper and a full rigorous treatment will appear in a forthcoming work.

6.3.2 Necessary Conditions of the Optimal Trajectories

Fix $x \in \mathcal{X}$ and $z \in \mathcal{Z}$ as initial conditions and fix the terminal time t . Denote by $\bar{\gamma}(s)$ and $\bar{\xi}(s)$ as the optimal state trajectories such that

$$\bar{\gamma}(s) := \bar{\gamma}(s; \chi) = \gamma(s; x, \bar{u}(\cdot; \chi)),$$

and

$$\bar{\xi}(s) := \bar{\xi}(s; \chi) = \xi(s; x, z, \bar{u}(\cdot; \chi)),$$

such that \bar{u} optimizes (6.2.12). By Pontryagin's theorem [79] there exists adjoint trajectories $p(s) := p(s; \chi)$ and $\lambda(s) := \lambda(s; \chi)$ such that the function

$$[0, t] \ni s \mapsto (\bar{\gamma}(s), \bar{\xi}(s), p(s), \lambda(s)) \tag{6.3.9}$$

is a solution of the characteristic system

$$\left\{ \begin{array}{l} \dot{\bar{\gamma}}(s) = \mathcal{H}_p(\bar{\gamma}(s), \bar{\xi}(s), p(s), \lambda(s)), \\ \dot{\bar{\xi}}(s) = \mathcal{H}_\lambda(\bar{\gamma}(s), \bar{\xi}(s), p(s), \lambda(s)), \\ \dot{p}(s) = -\mathcal{H}_x(\bar{\gamma}(s), \bar{\xi}(s), p(s), \lambda(s)), \\ \dot{\lambda}(s) = -\mathcal{H}_z(\bar{\gamma}(s), \bar{\xi}(s), p(s), \lambda(s)), \end{array} \right. \tag{6.3.10}$$

almost everywhere $s \in [0, t]$ with boundary conditions

$$p(t) = \mathbf{0}, \quad \lambda(t) = G_z(\bar{\xi}(t)).$$

6.3.3 Numerical Approximations Viscosity Solutions to First-Order Hyperbolic PDEs

Traditional methods for computing the viscosity solution to (6.3.5) rely on constructing a discrete grid of points. This is typically chosen as a Cartesian grid, but many other grid types exist. The value function is found using a *method of lines* (MOL) approach by the solving the following family of ODEs, pointwise at each grid point $\chi^k = (x^k, z^k) \in \mathcal{S} := \mathcal{X} \times \mathcal{Z}$:

$$\begin{cases} \dot{\phi}(s, \chi^k) = -\mathcal{H}(\chi^k, D_{\chi}\phi(s, \chi^k)), & s \in [0, t] \\ \phi(0, \chi^k) = G(z^k), \end{cases} \quad (6.3.11)$$

where $\phi(s, \chi^k)$ should be viewed as an approximation to the value function $V(s, \chi^k)$ in (6.3.5) and

$$D_{\chi}\phi(s, \chi^k) \approx \phi_{\chi}(s, \chi^k)$$

is obtained by a finite difference scheme used to approximate the gradient of ϕ at grid point k . Care must be taken when evaluating finite differences of possibly non-smooth functions and the family of *Essentially Non-Oscillatory* (ENO) methods were developed to address this issue [111]. The advantage of the method of lines is that we can compute (6.3.11) independently at each grid point with $\phi(t, \chi^k) \approx V(t, \chi^k)$. Under certain conditions, for example the Lax-Richtmyer equivalence theorem [112],

$$\Delta s \rightarrow 0, \Delta \chi \rightarrow 0 \implies \phi(t, \chi^k) \rightarrow V(t, \chi^k)$$

when the scheme is both consistent, i.e. the error between $\phi(t, \chi^k)$ and $V(t, \chi^k)$ tends to zero, and stable. In this case, stability is enforced when the time step, Δs , satisfies the

Courant-Friedrichs-Lewy (CFL) condition [113]. When the HJ equation is a non-linear PDE, then additionally a Lax-Friedrichs approximation [114, 77] is needed to ensure stability. In the Lax-Friedrichs method the Hamiltonian in (6.3.11) is replaced by

$$\hat{\mathcal{H}}(\chi, \sigma^+, \sigma^-) := \mathcal{H}\left(\chi, \frac{\sigma^+ + \sigma^-}{2}\right) - \nu(\chi)^\top \left(\frac{\sigma^+ + \sigma^-}{2}\right),$$

where inputs $D_\chi^+ \phi(s, \chi^k) \rightarrow \sigma^+$ and $D_\chi^- \phi(s, \chi^k) \rightarrow \sigma^-$ are the right and left side bias finite differencing approximations to the gradient, respectively. The term $\nu(\chi)$ is the artificial dissipation and depends on $H_\sigma(\chi, \sigma)$, the gradient of the Hamiltonian with respect to the adjoint variable. The MOL approach in (6.3.11) becomes

$$\begin{cases} \dot{\phi}(s, \chi^k) = -\hat{\mathcal{H}}(\chi^k, D_\chi^+ \phi(s, \chi^k), D_\chi^- \phi(s, \chi^k)), \\ \phi(0, \chi^k) = G(z^k), \end{cases} \quad (6.3.12)$$

In general, no closed form solution exists to (6.3.12) and therefore an explicit Runge-Kutta scheme is employed. If the first order Euler method is used to solve (6.3.12), then we have the following time-marching scheme with iteration for $s \in [0, t]$:

$$\begin{cases} \phi(s + \Delta s, \chi^k) = \phi(s, \chi^k) \\ \quad - \Delta s \hat{\mathcal{H}}(\chi^k, D_\chi^+ \phi(s, \chi^k), D_\chi^- \phi(s, \chi^k)), \\ \phi(0, \chi^k) = G(z^k). \end{cases} \quad (6.3.13)$$

The reader is encouraged to read [101] for a comprehensive review on numeric numeric methods to solving first-order hyperbolic HJ PDEs.

6.4 HJB Decomposition

We are especially interested in problems for which the x -component of the state in (6.3.1) has a relatively small dimension, but z -component does not. This is common in the vehicle sensing problem discussed in Section 6.2, because the dimension of z scales with the square of the number of parameters to be estimated and therefore, even for simple vehicle dynamics and a relatively small number of parameters, the dimension of the state χ is too large to apply (6.3.13). To overcome this challenge, we present an hybrid method of lines that uses a grid over x , but no grid over z .

A key challenge to creating such a method is to find a closed-form expression for the gradient of the value function with respect to z , so as to avoid finite differencing schemes in z . Taking advantage of the specific structure of the problem, we show that we can use a grid over the state variable x to compute $D_x \phi(s, \chi^k) \approx \phi_x(s, \chi^k)$ with finite differences, but avoid a grid over the state variable z by solving a family of ODEs to compute $D_z \phi(s, \chi^k)$. This is supported by the following theorem.

Theorem 6.1. *Suppose the value function $V(s, \chi)$ is twice differentiable at $(s, \chi) \in [0, \infty) \times \mathcal{S}$. Then at any point χ , the gradient of the value function with respect to z can be found using the following ODE:*

$$\begin{cases} \dot{V}_z(s, \chi) = -\frac{\partial}{\partial z} \langle G_z(\bar{\xi}(s)), \ell(x) \rangle \\ \quad -R_x(s, \chi, \pi(s, \chi), f(x), g(x)), \\ V_z(0, \chi) = G_z(z), \end{cases} \quad (6.4.1)$$

where

$$R_x(s, \chi, u, \alpha, \beta) := \frac{\partial}{\partial x} \left\{ \langle G_z(\bar{\xi}(s)), \alpha \rangle \right. \quad (6.4.2)$$

$$\left. + \langle G_z(\bar{\xi}(s)), \beta u \rangle \right\}. \quad (6.4.3)$$

The proof of Theorem 6.1 will need the following technical lemma.

Lemma 6.1. *Suppose that the gradient $V_z(t, \chi)$ exists at $(t, \chi) \in [0, \infty) \times \mathcal{S}$. Then the gradient of the value function with respect to the augmented variable is given by*

$$V_z(t, \chi) = G_z(\bar{\xi}(t; \chi)).$$

Proof. Recall from (6.3.4) and applying the optimal control sequence,

$$\bar{\xi}(s) = z + \int_0^s \ell(\bar{\gamma}(\tau)) d\tau.$$

Therefore

$$G_z\left(z + \int_0^t \ell(\bar{\gamma}(\tau)) d\tau\right) = G_z(\bar{\xi}(t)) := \lambda(t). \quad (6.4.4)$$

Recognize that (6.4.4) is the boundary condition of the characteristic system (6.3.10), and that

$$\begin{aligned} V_z(t, \chi) &= \lambda(0) \\ &= G_z(\bar{\xi}(t)) - \int_t^0 \mathcal{H}_z(\bar{\gamma}(s), \bar{\xi}(s), p(s), \lambda(s)) ds. \end{aligned}$$

Where the first line above uses the connection between the adjoint variable, λ , and the value function [109, Theorem 3.4, p. 235]. Observing that the Hamiltonian (6.3.8) does

not depend on the argument z , then it follows that

$$\mathcal{H}_z(\bar{\gamma}(s), \bar{\xi}(s), p(s), \lambda(s)) = 0, \quad s \in [0, t],$$

which leads to

$$V_z(t, \chi) = G_z(\bar{\xi}(t)).$$

□

We now proceed to the proof of Theorem 6.1.

Proof. Fix x, z and noting the original HJB equation (6.3.5):

$$\begin{aligned} \dot{V}_z(s, \chi) &= \frac{\partial}{\partial s} \{V_z(s, \chi)\} \\ &= \frac{\partial}{\partial z} \{V_s(s, \chi)\} \\ &= \frac{\partial}{\partial z} \{-\mathcal{H}(\chi, V_x(s, \chi), V_z(s, \chi))\}. \end{aligned}$$

From the definition of the Hamiltonian

$$\dot{V}_z(s, \chi) = \frac{\partial}{\partial z} \left\{ -\langle V_z(s, \chi), \ell(x) \rangle - \langle V_x(s, \chi), f(x) \rangle - \min_{u \in U} \langle V_x(s, \chi), g(x)u \rangle \right\}.$$

Fix time $s \in [0, t]$, and define the function

$$\varphi^s(\chi, u) := \min_{u \in U} F^s(\chi, u),$$

where

$$F^s(\chi, u) := \langle V_x(s, \chi), g(x)u \rangle,$$

and recall that

$$\pi(s, \chi) := \arg \min_{u \in U} \langle V_x(s, \chi), g(x) u \rangle.$$

Since by assumption both $V_x(s, \chi)$ and $V_{zx}(s, \chi)$ exist, and $F^s(\chi, u)$ is differentiable at χ , this implies the gradient of φ^s can be found [115, Theorem 4.13] with the following relation:

$$\varphi_z^s(\chi, u) = F_z^s(\chi, \pi(s, \chi)).$$

This gives

$$\begin{aligned} \dot{V}_z(s, \chi) = & - \frac{\partial}{\partial z} \{ \langle V_z(s, \chi), \ell(x) \rangle \} \\ & - \frac{\partial}{\partial z} \{ \langle V_x(s, \chi), \alpha \rangle \} \Big|_{\alpha=f(x)} \\ & - \frac{\partial}{\partial z} \{ \langle V_x(s, \chi), \beta u \rangle \} \Big|_{u=\pi(s, \chi), \beta=g(x)}. \end{aligned}$$

Noting the symmetry of the gradients with respect to x, z we have

$$\begin{aligned} \dot{V}_z(s, \chi) = & - \frac{\partial}{\partial z} \{ \langle V_z(s, \chi), \ell(x) \rangle \} \\ & - \frac{\partial}{\partial x} \{ \langle V_z(s, \chi), \alpha \rangle \} \Big|_{\alpha=f(x)} \\ & - \frac{\partial}{\partial x} \{ \langle V_z(s, \chi), \beta u \rangle \} \Big|_{u=\pi(s, \chi), \beta=g(x)}, \end{aligned}$$

and then applying Lemma 6.1, the result follows. \square

6.4.1 Method of Lines with State Space Decomposition

Recall that we denote by $\phi(s, \chi)$ the numeric approximation to the value function, $V(s, \chi)$. The proposed hybrid MOL relies on an approximation $D_x \phi(s, \chi)$ of the gradient of the value function with respect to x , $V_x(s, \chi)$, that is based on the Lax-

Friedrichs approximation. However, the approximation $\Phi(s, \chi)$ of the gradient of the value function with respect to z , $V_z(s, \chi)$, is obtained by solving an ODE in time and does not require a spatial grid. In view of this, this method computes the two approximations $\phi(s, x^k, z)$ and $\Phi(s, x^k, z)$ on points $(x^k, z) \in \mathcal{S}$ where the x^k are restricted to a finite grid of the x -component of the state, whereas z is not restricted to a grid. To accomplish this, we need the following assumption that, together with Theorem 1, leads to the following MOL.

Suppose that the first term in (6.4.1) can be written as

$$\frac{\partial}{\partial z} \{ \langle G_z(\bar{\xi}(s)), \ell(x) \rangle \} = \Upsilon(x, z, G_z(\bar{\xi}(s))), \quad (6.4.5)$$

and fix z for any $z \in \mathcal{Z}$. Denote by $\Phi(s, x^k, z) \approx \phi_z(s, x^k, z) = G_z(\bar{\xi}(s))$ as the gradient estimate of the value function with respect to z . Then from Theorem 6.1 and Lemma 6.1, we construct the following method of lines approach, for $(x^k, z) \in \mathcal{S}$:

$$\left\{ \begin{array}{l} \dot{\phi}(s, x^k, z) = -\tilde{\mathcal{H}}(x^k, z, D_x^+ \phi(s, x^k, z), D_x^- \phi(s, x^k, z), \\ \quad \Phi(s, x^k, z)), \\ \dot{\Phi}(s, x^k, z) = -\Upsilon(x^k, z, \Phi(s, x^k, z)) \\ \quad -R_x(s, x^k, z, \pi(s, x^k, z), f(x^k), g(x^k)), \\ \phi(0, x^k, z) = G(z), \\ \Phi(0, x^k, z) = G_z(z), \end{array} \right. \quad (6.4.6)$$

where

$$\begin{aligned} \tilde{\mathcal{H}}(x, z, \rho^+, \rho^-, \lambda) := & \mathcal{H}\left(x, z, \frac{\rho^+ + \rho^-}{2}, \lambda\right) \\ & - \nu(x)^\top \left(\frac{\rho^+ + \rho^-}{2}\right), \end{aligned}$$

is the Lax-Friedrichs approximation. The Lax-Friedrichs approximation is only needed in the x dimension since that is the only space where a grid is constructed for computing finite differences.

6.5 Optimal Information Collection

Recall that the system (6.2.8) presented in Section 6.2 is of the form of Section 6.3, and we can use Theorem 6.1 to construct a method of lines. Recall that for Dubins car, $U = [-\omega_{\max}, \omega_{\max}]$, and the optimal Hamilton (6.3.6) becomes

$$\mathcal{H}(x, z, p, \lambda) = \langle f(x), p \rangle + \omega_{\max} |B^\top p| + \langle \lambda, \text{vec}(Q(x)) \rangle,$$

and optimal control policy is given by

$$\begin{aligned} \pi(s; x, z) := & \arg \min_{u \in U} H(x, z, u, V_x(s, x, z), V_z(s, x, z)) \\ & \in \begin{cases} -\omega_{\max} & B^\top V_x(s, x, z) < 0 \\ [-\omega_{\max}, \omega_{\max}] & B^\top V_x(s, x, z) = 0 \\ \omega_{\max} & B^\top V_x(s, x, z) > 0. \end{cases} \end{aligned} \quad (6.5.1)$$

In order to compute the first term in (6.4.1) for the vehicle tracking problem presented in Section 6.2, we present the following lemma.

Lemma 6.2. *Let $\chi \in \mathcal{S}$. When $G(z) = -\log \det(\text{vec}^{-1}(z))$ and $\ell(x) = \text{vec}(Q(x))$, then*

$$\begin{aligned} & \frac{\partial}{\partial z} \langle G_z(\bar{\xi}(s)), \ell(x) \rangle \\ &= \text{vec}(\text{vec}^{-1}(G_z(\bar{\xi}(s))) \cdot Q(x) \cdot \text{vec}^{-1}(G_z(\bar{\xi}(s))))^\top. \end{aligned}$$

Proof. Define $\bar{\Xi}(z) := \Xi(s; x, z, \bar{u}(\cdot)) = \text{vec}^{-1}(\bar{\xi}(s; \chi, \bar{u}(\cdot)))$ as the optimal auxiliary state trajectory at the time, s , reshaped into a matrix. The matrix forms simplifies the following proof and the computations in the examples to follow. We also denote by $Z := \text{vec}^{-1}(z)$. The gradient with respect to a matrix of a function $F(Z)$ is the matrix defined by

$$\frac{\partial}{\partial Z} F(Z) := \text{vec}^{-1} \left\{ \left[\frac{\partial F(Z)}{\partial Z_{ij}} \right]_{i,j} \right\}.$$

Recall (6.2.11) and from Lemma 6.1 that $V_z(s, x, z) = G_z(\bar{\Xi}(z)) = \text{vec}^{-1}(\bar{\Xi}(z)^{-1})$.

Then we have

$$\begin{aligned} & \frac{\partial}{\partial z} \langle G_z(\bar{\xi}(s)), \ell(x) \rangle \\ &= \text{vec} \left(\frac{\partial}{\partial Z} \text{tr}(\bar{\Xi}(z)^{-1} Q(x)) \right). \end{aligned}$$

We direct our attention to the term inside the vec operator in the last line above, and find

$$\begin{aligned} & \frac{\partial}{\partial Z_{ij}} \text{tr}(\bar{\Xi}(z)^{-1} Q(x)) \\ &= \text{tr} \left(\frac{\partial}{\partial Z_{ij}} \{ \bar{\Xi}(z)^{-1} \} Q(x) \right) \\ &= \text{tr} \left(-\bar{\Xi}(z)^{-1} \frac{\partial \bar{\Xi}(z)}{\partial Z_{ij}} \bar{\Xi}(z)^{-1} Q(x) \right) \end{aligned}$$

where the last line is from [116]. Noting $\frac{\partial \bar{\Xi}(z)}{\partial Z_{ij}} = \frac{\partial \bar{\Xi}(z)}{\partial Z} \frac{\partial z}{\partial Z_{ij}}$, recalling from Proposition

B.1 that $\frac{\partial \bar{\Xi}(z)}{\partial Z} = I$ and noting $\frac{\partial z}{\partial Z_{ij}} = S^{ij} := e_i e_j^\top$, where e_k is a vector with a 1 in k -th element and zeros elsewhere. We now have

$$\begin{aligned}
& \frac{\partial}{\partial Z_{ij}} \text{tr} \left(\bar{\Xi}(z)^{-1} Q(x) \right) \\
&= -\text{tr} \left(\bar{\Xi}(z)^{-1} e_i e_j^\top \bar{\Xi}(z)^{-1} Q(x) \right) \\
&= -\text{tr} \left(e_j^\top \bar{\Xi}(z)^{-1} Q(x) \bar{\Xi}(z)^{-1} e_i \right) \\
&= -e_j^\top \bar{\Xi}(z)^{-1} Q(x) \bar{\Xi}(z)^{-1} e_i \\
&= \left[\bar{\Xi}(z)^{-1} Q(x) \bar{\Xi}(z)^{-1} \right]_{ji}, \\
&= \left[\text{vec}^{-1} \left(G_z \left(\bar{\xi}(s) \right) \right) \cdot Q(x) \cdot \text{vec}^{-1} \left(G_z \left(\bar{\xi}(s) \right) \right) \right]_{ji}
\end{aligned}$$

and the result follows. \square

Note Lemma 6.2 gives us the relation in (6.4.5) for the sensing trajectory problem, and when in matrix form as in the proof, gives a relationship that is simple to compute.

6.6 Results

We consider a passive RF sensor that measures the Doppler frequency shift in the carrier frequency, denoted as \mathcal{F} , arising from the relative motion between transmitting vehicle and the receiver. Note that we do not need to decode the underlying transmission, as we are only tracking the shifts in received carrier frequency. More details about the derivation of this, as well as other sensor models can be found in [100].

We assume in this paper the sensor produces conditionally independent measurements, each with a Gaussian distribution with mean $\mu_{\mathcal{F}}(\theta)$. While the mean vector depends on the parameter of interest, θ , the covariance does not depend¹ on θ and is

¹It is not required that the covariance to be independent of θ , but it simplifies the example here.

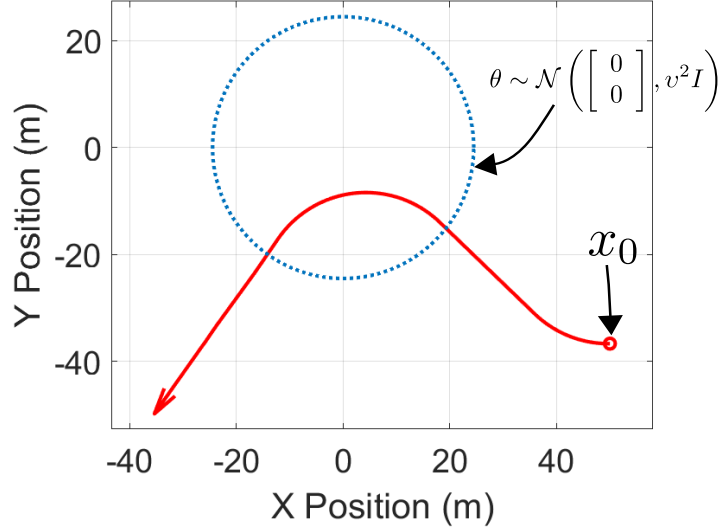


Figure 6.6.1: An optimal path computed for the first example, shown in red. In this example, the aircraft is only using Doppler shift measurements. The blue dashed circle is the 95% error ellipse of the prior distribution on θ , which in this example represents the position of the vehicle target.

given as $\Sigma_{\mathcal{F}}$. This gives a closed form expression for (6.2.5) for measurement \mathcal{F} , as

$$Q(x; \theta) = \left(\frac{\partial \mu_{\mathcal{F}}(\theta)}{\partial \theta} \right)^{\top} \Sigma_{\mathcal{F}}^{-1} \left(\frac{\partial \mu_{\mathcal{F}}(\theta)}{\partial \theta} \right), \quad (6.6.1)$$

where $\frac{\partial \mu_{\mathcal{F}}(\theta)}{\partial \theta}$ denotes the Jacobian matrix of $\mu_{\mathcal{F}}(\theta)$ [95]. To estimate the expectation and find the expression (6.2.4), we choose a second-order Taylor series expansion. Let $Q_{ij}(x; \theta)$ denote the i, j -th element of the (6.6.1), and θ is a random variable with mean μ_{θ} and covariance Σ_{θ} . Then we approximate the element with a second order Taylor expansion as

$$\begin{aligned} Q_{ij}(x; \theta) &\approx Q_{ij}(x; \mu_{\theta}) + \nabla Q_{ij}(x; \mu_{\theta})^{\top} (\theta - \mu_{\theta}) \\ &\quad + \frac{1}{2} (\theta - \mu_{\theta})^{\top} \mathbf{H}_{ij}(x; \mu_{\theta}) (\theta - \mu_{\theta}), \end{aligned}$$

where $\mathbf{H}_{ij}(x; \theta)$ is the hessian matrix of $Q_{ij}(x; \theta)$ with respect to θ . The expected value

is then found as

$$\mathbb{E}_\theta [Q_{ij}(x; \theta)] \approx Q_{ij}(x; \mu_\theta) + \frac{1}{2} \text{tr}(\Sigma_\theta \mathbf{H}_{ij}(x; \mu_\theta)). \quad (6.6.2)$$

The closed-form gradient $\frac{\partial \mu_{\mathcal{F}}(\theta)}{\partial \theta}$ in (6.6.1) are found from [100], while the Hessian values were found using the CASADI toolbox [117].

In the example the parameters to be estimated, θ , consist of the $(X, Y) \in \mathbb{R}^2$ position of the target vehicle. The prior distribution of θ is given as

$$\theta \sim \mathcal{N} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, v^2 I \right),$$

where $v = 10m$ is the standard deviation. The sensor measures the Doppler shifts with noise standard deviation of $\Sigma_{\mathcal{F}} = 1$. The sensing aircraft is flying $1000m$ above the ground level where the target vehicle is located and the turn rate is limited with $\omega_{\max} = 0.05$ rad/s.

Figure 6.6.1 shows the optimal path from the initial condition of $X(0) = 50m$, $Y(0) = -36.6m$, and $\psi(0) = -\pi$. The initial angle of $-\pi$ implies the tracking aircraft is moving from right to left initially at $t = 0$. It can be seen in the figure that the optimal path begins with turning maneuvers before traveling straight along a ray extending outward from the center of the prior distribution of θ . Conceptually, travel along this ray will give maximum variation in Doppler shift, but the early maneuvers are still necessary since multiple directions of measurements are required to fully localize using only Doppler measurements. Figure 6.6.2 shows a series of optimal paths generated with same initial conditions for $X(0)$ and $\psi(0)$, but with a variation in the initial condition, $Y(0)$. The vertical initial condition, $Y(0)$, were chosen uniformly from a range $[-50, 50]$. While the trajectories are different quantitatively from that of Figure 6.6.1, they share the same

qualitative properties of an initial maneuver to gain measurements in various directions before traveling away from the prior belief, on a ray extending directly from the center.

6.7 Conclusion

We present a hybrid method of lines approach for solving a class of Hamilton–Jacobi PDEs that arise in the optimal placement of sensors. This method provides for robustness, where needed, in the x subspace by using a classic grid approach with finite differencing. It avoids a grid in the z subspace and hence scales well with the number of z dimensions. We applied this to a trajectory optimization problem where the goal is to find the trajectory that minimizes the estimation error from the measurements collected along the calculated path. Future work includes investigating metrics other than LOGDET such as the trace of the inverse and studying if the hybrid method of lines approach can be generalized to a broader class of systems.

Acknowledgments

The authors would like to thank Levon Nurbekyan, with the Department of Mathematics at UCLA, for providing a reference that assisted in the proof of Theorem 6.1.

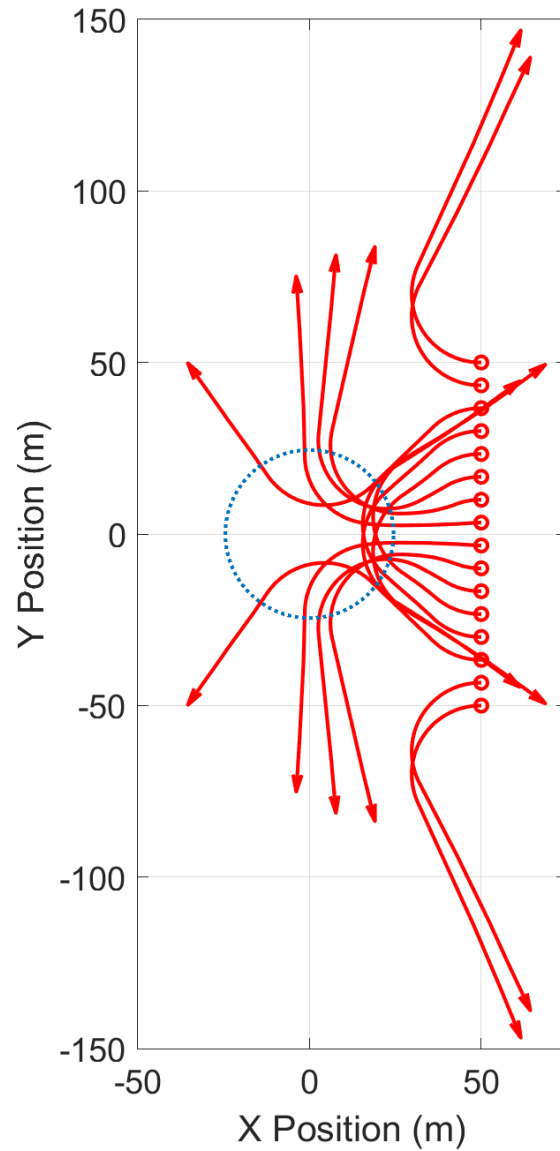


Figure 6.6.2: Here a series of optimal trajectories are shown in red from different starting locations, with each vehicle starting out moving from right to left. Same as in Fig. 6.6.1, the aircraft is only using Doppler shift measurements. The blue circle is the 95% error ellipse of the prior distribution on θ .

Appendix A

Appendix for Chapter 4

Lemma A.1. *Given that (4.3.4) satisfies assumptions 1-3, then (4.4.3) also satisfies assumption 1-3.*

Proof. From assumption 1, \widehat{H}_i is continuous by composition rule [76, Theorem 4.7] and since (4.4.6) does not depend on state, it therefore trivially meets assumption 3. It remains to show that (4.4.3) meets assumption 2. We know assumption 2 holds for all $x_i \in \mathbb{R}^{n_i}$ and for all $p', p'' \in \mathbb{R}^{n_i}$, therefore the following holds

$$\begin{aligned} & \left| H_i \left(s, x_i, e^{-sA_i^\top} p' \right) - H_i \left(s, x_i, e^{-sA_i^\top} p'' \right) \right| \\ & \leq c_i (1 + \|x_i\|) \left\| e^{-sA_i^\top} p' - e^{-sA_i^\top} p'' \right\| \\ & \leq c_i (1 + \|x_i\|) \left| \lambda_{\max} \left(e^{-sA_i^\top} \right) \right| \|p' - p''\| \\ & = c (1 + \|x_i\|) \|p' - p''\|. \end{aligned}$$

When $x_i = 0$ we have

$$\begin{aligned} & \left| H_i \left(s, 0, e^{-sA_i^\top} p' \right) - H_i \left(s, 0, e^{-sA_i^\top} p'' \right) \right| \\ &= \left| \widehat{H}_i(s, p') - \widehat{H}_i(s, p'') \right| \\ &\leq c \|p' - p''\| \leq c(1 + \|x_i\|) \|p' - p''\|, \end{aligned}$$

for any x_i . And since $\left| \widehat{H}_i(s, x_i, 0) \right| = 0$ for all x_i , it follows that assumption 2 is met. \square

Proof of Lemma 4.1.

Proof. The sum of continuous functions is also continuous [76, Theorem 4.9], therefore assumption 1 is met. To prove assumption 2, we first write, for all $\bar{p}', \bar{p}'' \in \mathbb{R}^n$,

$$\begin{aligned} & |H(s, x, \bar{p}') - H(s, x, \bar{p}'')| \\ &= \left| \sum_i H_i(s, x_i, \bar{p}'_i) - \sum_i H_i(s, x_i, \bar{p}''_i) \right| \\ &\leq \sum_i |H_i(s, x_i, \bar{p}'_i) - H_i(s, x_i, \bar{p}''_i)|, \end{aligned} \tag{A.0.1}$$

$$\begin{aligned} &\leq \sum_i c_i (1 + \|x_i\|) \|\bar{p}'_i - \bar{p}''_i\| \\ &\leq \sum_i c_i (1 + \|x\|) \|\bar{p}' - \bar{p}''\| \end{aligned} \tag{A.0.2}$$

$$c(1 + \|x\|) \|\bar{p}' - \bar{p}''\|,$$

where line (A.0.1) comes from the triangular inequality and line (A.0.2) comes by noting $\forall i, \|x_i\| \leq \|x\|$ and $\|\bar{p}'_i - \bar{p}''_i\| \leq \|\bar{p}' - \bar{p}''\|$. Therefore, there exists a $c = \sum_i c_i$ such that the inequality holds and we arrive at our result for part 1 of assumption 2. The second

part of assumption follows from part 1,

$$\begin{aligned}
|H(s, x, 0)| &= \left| \sum_i H(s, x_i, 0) \right| \\
&\leq \sum_i |H(s, x_i, 0)| \leq \sum_i c_i (1 + \|x_i\|) \\
&\leq \sum_i c_i (1 + \|x\|) = c(1 + \|x\|).
\end{aligned}$$

And we have shown part 2. The proof of assumption 3 follows that of above. For any compact set $M \subset \mathbb{R}^n$ and for all $\bar{x}', \bar{x}'' \in M$

$$\begin{aligned}
&|H(s, \bar{x}', p) - H(s, \bar{x}'', p)| \\
&= \left| \sum_i H(s, \bar{x}'_i, p_i) - \sum_i H_i(s, \bar{x}''_i, p_i) \right| \\
&\leq \sum_i |H(s, \bar{x}'_i, p_i) - H_i(s, \bar{x}''_i, p_i)|, \\
&\leq \sum_i \kappa_i(M) (1 + \|p_i\|) \|\bar{x}'_i - \bar{x}''_i\| \\
&\leq \sum_i \kappa_i(M) (1 + \|p\|) \|\bar{x}' - \bar{x}''\| \\
&= \kappa(M) (1 + \|p\|) \|\bar{x}' - \bar{x}''\|.
\end{aligned}$$

Therefore, there exists a $\kappa(M) = \sum_i \kappa_i(M)$ such that the inequality holds. \square

Appendix B

Appendix for Chapter 6

B.0.1 Regularity Assumptions of the Hamiltonian

Let n be the dimension of the augmented state variable χ , and denote by $\sigma := (p, \lambda)^\top$, and with a slight abuse of notation note that $\mathcal{H}(s, \chi, \sigma) = \mathcal{H}(s, x, z, \sigma) = \mathcal{H}(s, x, z, p, \lambda)$ and vice versa. We introduce a set of mild regularity assumptions:

(H1) The Hamiltonian

$$\begin{aligned} [0, t] \times \mathcal{X} \times \mathcal{Z} \times \mathbb{R}^n \ni (s, x, z, p, \lambda) \\ \mapsto \mathcal{H}(s, x, z, p, \lambda) \in \mathbb{R} \end{aligned}$$

is continuous.

(H2) There exists a constant $c > 0$ such that for all $(s, x, z) \in [0, t] \times \mathcal{X} \times \mathcal{Z}$ and for all $\sigma', \sigma'' \in \mathbb{R}^n$, the following inequalities hold

$$\begin{aligned} |\mathcal{H}(s, x, z, \sigma') - \mathcal{H}(s, x, z, \sigma'')| \leq \kappa_1(\chi) \\ \times \|\sigma' - \sigma''\|, \end{aligned}$$

and

$$|\mathcal{H}(s, x, z, \mathbf{0})| \leq \kappa_1(\chi),$$

with $\kappa_1(\chi) = c(1 + \|\chi\|)$.

(H3) For any compact set $M \subset \mathbb{R}^n$ there exists a constant $C(M) > 0$ such that for all $\chi', \chi'' \in M$ and for all $(s, \sigma) \in [0, t] \times \mathbb{R}^n$ the inequality holds

$$|\mathcal{H}(s, \chi', \sigma) - \mathcal{H}(s, \chi'', \sigma)| \leq \kappa_2(\sigma) \|\chi' - \chi''\|,$$

with $\kappa_2(\sigma) = C(M)(1 + \|\sigma\|)$.

(H4) The terminal cost function

$$\mathbb{R}^n \ni \chi \mapsto G(\chi) \in \mathbb{R},$$

is continuous.

Next we present an important theorem on the existence and uniqueness of viscosity solutions of the Hamilton–Jacobi equation.

Theorem B.1 ([75, Theorem II.8.1, p. 70]). *Let assumptions (H1) – (H4) hold. Then there exists a unique viscosity solution to (6.3.5).*

B.0.2 Supporting Propositions

Proposition B.1. *Let $\chi \in \mathcal{S}$, then*

$$\frac{\partial}{\partial z} \xi(t; \chi, \bar{u}(\cdot)) = I.$$

Proof. By assumption, the terminal point of the state trajectory $\zeta(t; \chi, \bar{u}(\cdot))$ is differentiable with respect to initial condition $\chi \in \mathcal{S}$. Defining the Jacobin, for $s \in [0, t]$,

$$\begin{aligned} m(s) &:= \begin{bmatrix} m_{xx}(s) & m_{x,z}(s) \\ m_{zx}(s) & m_{zz}(s) \end{bmatrix} \\ &= \begin{bmatrix} \bar{\gamma}_x(s) & \bar{\gamma}_z(s) \\ \bar{\xi}_x(s) & \bar{\xi}_z(s) \end{bmatrix} = \frac{\partial}{\partial \chi} \zeta(s; \chi, \bar{u}(\cdot)). \end{aligned}$$

We have from [109, Chapter 5, Equation 3.23] that $m(t)$ satisfies the following matrix equation almost everywhere:

$$\begin{cases} \dot{m}(s) = \hat{f}_\chi(\bar{\zeta}(s; \chi, \bar{u}(\cdot)), \bar{u}(s)) m(s), & s \in [0, t], \\ m(0) = I. \end{cases}$$

From which the m_{zz} partition is written as

$$\begin{cases} \dot{m}_{zz}(s) = \ell_z(\bar{\gamma}(s; \chi, \bar{u}(\cdot))) m_{zz}(s), & s \in [0, t], \\ m_{zz}(0) = I. \end{cases}$$

Since ℓ does not depend on z , we have

$$\dot{m}_{zz}(s) = 0, \forall s \in [0, t],$$

and the result follows. □

Bibliography

- [1] D. Grimsman, M. R. Kirchner, J. P. Hespanha, and J. R. Marden, *The impact of measurement passing in sensor network measurement selection*, *IEEE Transactions on Control of Network Systems* **10** (2023), no. 1 112–123.
- [2] M. R. Kirchner, D. Grimsman, J. P. Hespanha, and J. R. Marden, *Trajectories for the optimal collection of information*, in *2023 IEEE Aerospace Conference*, pp. 1–10, 2023.
- [3] D. Grimsman, M. R. Kirchner, J. P. Hespanha, and J. R. Marden, *The impact of message passing in agent-based submodular maximization*, in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 530–535, IEEE, 2020.
- [4] N. P. Osmolovskii, *Calculus of Variations and Optimal Control*, vol. 180. American Mathematical Society, 1998.
- [5] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, vol. 153. Springer Science & Business Media, 2006.
- [6] I. M. Mitchell, *The flexible, extensible and efficient toolbox of level set methods*, *Journal of Scientific Computing* **35** (2008), no. 2 300–329.
- [7] I. Mitchell, A. M. Bayen, and C. J. Tomlin, *A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games*, *IEEE Transactions on Automatic Control* **50** (2005), no. 7 947–957.
- [8] R. E. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 2015.
- [9] R. E. Bellman, *Dynamic Programming*, vol. 1. Princeton University Press, 1957.
- [10] J. Darbon and S. Osher, *Algorithms for overcoming the curse of dimensionality for certain Hamilton-Jacobi equations arising in control theory and elsewhere*, *Research in the Mathematical Sciences* **3** (2016), no. 1 19.
- [11] M. R. Kirchner, R. Mar, G. Hewer, J. Darbon, S. Osher, and Y. T. Chow, *Time-optimal collaborative guidance using the generalized Hopf formula*, *IEEE Control Systems Letters* **2** (2018), no. 2 201–206.

- [12] N. F. Palumbo, R. A. Blauwkamp, and J. M. Lloyd, *Modern homing missile guidance theory and techniques*, *Johns Hopkins APL Technical Digest* **29** (2010), no. 1 42–59.
- [13] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, *SIAM Journal on Imaging Sciences* **2** (2009), no. 2 323–343.
- [14] Y. T. Chow, J. Darbon, S. Osher, and W. Yin, *Algorithm for overcoming the curse of dimensionality for time-dependent non-convex Hamilton–Jacobi equations arising from optimal control and differential games problems*, *Journal of Scientific Computing* (2016) 1–27.
- [15] P. L. Combettes and J.-C. Pesquet, *Proximal splitting methods in signal processing*, in *Fixed-point Algorithms for Inverse Problems in Science and Engineering*, pp. 185–212. Springer, 2011.
- [16] N. Parikh and S. Boyd, *Proximal algorithms*, *Foundations and Trends in Optimization* **1** (2014), no. 3 127–239.
- [17] A. R. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation and Control*. CRC Press, 1975.
- [18] L. C. Evans, *Partial Differential Equations*. American Mathematical Society, Providence, R.I., 2010.
- [19] E. Hopf, *Generalized solutions of non-linear equations of first order*, *Journal of Mathematics and Mechanics* **14** (1965) 951–973.
- [20] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of convex analysis*. Springer Science & Business Media, 2012.
- [21] A. B. Kurzhanski and P. Varaiya, *Dynamics and Control of Trajectory Tubes: Theory and Computation*, vol. 85. Springer, 2014.
- [22] I. M. Mitchell, *A toolbox of level set methods*, Tech. Rep. TR-2007-11, UBC Department of Computer Science, 2007.
- [23] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, *Journal of Mathematical Imaging and Vision* **40** (2011), no. 1 120–145.
- [24] J. H. Mathews and K. D. Fink, *Numerical Methods using MATLAB*, vol. 3. Prentice Hall, 1999.
- [25] I. M. Ross and M. Karpenko, *A review of pseudospectral optimal control: From theory to flight*, *Annual Reviews in Control* **36** (2012), no. 2 182–197.

- [26] T. Pock and A. Chambolle, *Diagonal preconditioning for first order primal-dual algorithms in convex optimization*, in *2011 IEEE International Conference on Computer Vision (ICCV)*, pp. 1762–1769, IEEE, 2011.
- [27] A. H. Al-Mohy and N. J. Higham, *Computing the action of the matrix exponential, with an application to exponential integrators*, *SIAM Journal on Scientific Computing* **33** (2011), no. 2 488–511.
- [28] T. Goldstein, M. Li, X. Yuan, E. Esser, and R. Baraniuk, *Adaptive primal-dual hybrid gradient methods for saddle-point problems*, *arXiv preprint arXiv:1305.0546* (2013).
- [29] P. L. Lions and J.-C. Rochet, *Hopf formula and multitime Hamilton-Jacobi equations*, *Proceedings of the American Mathematical Society* **96** (1986), no. 1 79–84.
- [30] J. Darbon, *On convex finite-dimensional variational methods in imaging sciences and Hamilton-Jacobi equations*, *SIAM Journal on Imaging Sciences* **8** (2015), no. 4 2268–2293.
- [31] I. M. Ross, *A primer on Pontryagin’s principle in optimal control*. Collegiate Publishers, 2015.
- [32] I. M. Mitchell and J. A. Templeton, *A toolbox of Hamilton-Jacobi solvers for analysis of nondeterministic continuous and hybrid systems*, in *HSCC*, vol. 5, pp. 480–494, Springer, 2005.
- [33] J.-P. Aubin, L. Chen, O. Dordan, A. Faleh, G. Lezan, and F. Planchet, *Stochastic and tyochastic approaches to guaranteed ALM problem*, *Bulletin Français d’Actuariat* **12** (2012), no. 23 59–95.
- [34] S. J. Julier, *The scaled unscented transformation*, in *American Control Conference, 2002. Proceedings of the 2002*, vol. 6, pp. 4555–4559, IEEE, 2002.
- [35] S. J. Julier and J. K. Uhlmann, *Unscented filtering and nonlinear estimation*, *Proceedings of the IEEE* **92** (2004), no. 3 401–422.
- [36] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT press, 2005.
- [37] I. M. Ross, R. J. Proulx, and M. Karpenko, *Unscented guidance*, in *American Control Conference (ACC), 2015*, pp. 5605–5610, IEEE, 2015.
- [38] I. M. Ross, M. Karpenko, and R. J. Proulx, *Path constraints in tyochastic and unscented optimal control: Theory, application and experimental results*, in *American Control Conference (ACC), 2016*, pp. 2918–2923, IEEE, 2016.

- [39] N. F. Palumbo, R. A. Blauwkamp, and J. M. Lloyd, *Basic principles of homing guidance*, *Johns Hopkins APL Technical Digest* **29** (2010), no. 1 25–41.
- [40] R. Isaacs, *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*. Courier Corporation, 1999.
- [41] S. Pan, H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, *Pursuit, evasion and defense in the plane*, in *American Control Conference (ACC), 2012*, pp. 4167–4173, IEEE, 2012.
- [42] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, *A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag*, in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 1451–1456, IEEE, 2011.
- [43] D. M. Stipanović, G. Inalhan, R. Teo, and C. J. Tomlin, *Decentralized overlapping control of a formation of unmanned aerial vehicles*, *Automatica* **40** (2004), no. 8 1285–1296.
- [44] L. E. Dubins, *On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents*, *American Journal of Mathematics* **79** (1957), no. 3 497–516.
- [45] I. Ekeland and R. Temam, *Convex Analysis and Variational Problems*. SIAM, 1999.
- [46] L. C. Evans and P. E. Souganidis, *Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations.*, tech. rep., DTIC Document, 1983.
- [47] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [48] H. Anton, S. Davis, and I. Bivens, *Calculus: A New Horizon*. Wiley New York, 1999.
- [49] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, vol. 2. McGraw-Hill New York, 1998.
- [50] M. Akian, R. Bapat, and S. Gaubert, *Max-plus algebra*, *Handbook of Linear Algebra (Discrete Mathematics and its Applications)* **39** (2006) 10–14.
- [51] W. H. Fleming, *Deterministic nonlinear filtering*, *Annali della Scuola Normale Superiore di Pisa-Classe di Scienze* **25** (1997), no. 3-4 435–454.
- [52] W. M. McEneaney, *Max-Plus Methods for Nonlinear Control and Estimation*. Springer Science & Business Media, 2006.

- [53] W. M. McEneaney and A. Pandey, *An idempotent algorithm for a class of network-disruption games*, *Kybernetika* **52** (2016), no. 5 666–695.
- [54] J.-J. Slotine and W. Li, *Applied Nonlinear Control*, vol. 199. Prentice-Hall Englewood Cliffs, NJ, 1991.
- [55] X. Dong, B. Yu, Z. Shi, and Y. Zhong, *Time-varying formation control for unmanned aerial vehicles: Theories and applications*, *IEEE Transactions on Control Systems Technology* **23** (2015), no. 1 340–348.
- [56] A. Felner, R. Stern, S. E. Shimony, E. Boyarski, M. Goldenberg, G. Sharon, N. Sturtevant, G. Wagner, and P. Surynek, *Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges*, in *Tenth Annual Symposium on Combinatorial Search*, 2017.
- [57] D. Mellinger, A. Kushleyev, and V. Kumar, *Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams*, in *2012 IEEE International Conference on Robotics and Automation*, pp. 477–483, IEEE, 2012.
- [58] F. Augugliaro, A. P. Schoellig, and R. D’Andrea, *Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach*, in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1917–1922, IEEE, 2012.
- [59] M. Chen, J. F. Fisac, S. Sastry, and C. J. Tomlin, *Safe sequential path planning of multi-vehicle systems via double-obstacle hamilton-jacobi-isaacs variational inequality*, in *2015 European Control Conference (ECC)*, pp. 3304–3309, IEEE, 2015.
- [60] M. Debord, W. Hönig, and N. Ayanian, *Trajectory planning for heterogeneous robot teams*, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7924–7931, IEEE, 2018.
- [61] D. R. Robinson, R. T. Mar, K. Estabridis, and G. Hoyer, *An efficient algorithm for optimal trajectory generation for heterogeneous multi-agent systems in non-convex environments*, *IEEE Robotics and Automation Letters* **3** (2018), no. 2 1215–1222.
- [62] D. P. Bertsekas, *Linear Network Optimization: Algorithms and Codes*. MIT Press, 1991.
- [63] J. R. Marden and A. Wierman, *Distributed welfare games with applications to sensor coverage*, in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pp. 1708–1713, IEEE, 2008.

- [64] G. Arslan, J. R. Marden, and J. S. Shamma, *Autonomous vehicle-target assignment: A game-theoretical formulation*, *Journal of Dynamic Systems, Measurement, and Control* **129** (2007), no. 5 584–596.
- [65] J. R. Marden and M. Effros, *The price of selfishness in network coding*, *IEEE Transactions on Information Theory* **58** (2012), no. 4 2349–2361.
- [66] T. Roughgarden, *Selfish Routing and the Price of Anarchy*, vol. 174. MIT Press Cambridge, 2005.
- [67] J. R. Marden and T. Roughgarden, *Generalized efficiency bounds in distributed resource allocation*, *IEEE Transactions on Automatic Control* **59** (2014), no. 3 571–584.
- [68] M. R. Kirchner, *Towards general performance bounds of the distributed sensor coverage problem*, *Electrical, Computer & Energy Engineering Graduate Theses & Dissertations* (2013), no. 71.
- [69] W. Hönig, S. Kiesel, A. Tinka, J. W. Durham, and N. Ayanian, *Conflict-based search with optimal task assignment*, in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 757–765, International Foundation for Autonomous Agents and Multiagent Systems, 2018.
- [70] M. Turpin, N. Michael, and V. Kumar, *CAPT: Concurrent assignment and planning of trajectories for multiple robots*, *The International Journal of Robotics Research* **33** (2014), no. 1 98–112.
- [71] D. Morgan, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, *Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming*, *The International Journal of Robotics Research* **35** (2016), no. 10 1261–1285.
- [72] O. Gross, *The bottleneck assignment problem*, tech. rep., RAND CORP SANTA MONICA CALIF, 1959.
- [73] M. R. Kirchner, G. Hoyer, J. Darbon, and S. Osher, *A primal-dual method for optimal control and trajectory generation in high-dimensional systems*, in *IEEE Conference on Control Technology and Applications*, pp. 1575–1582, 2018.
- [74] R. Burkard, M. Dell’Amico, and S. Martello, *Assignment Problems: Revised Reprint*. SIAM, 2012.
- [75] A. I. Subbotin, *Generalized Solutions of First Order PDEs: The Dynamical Optimization Perspective*. Birkhäuser, 1995.
- [76] W. Rudin, *Principles of Mathematical Analysis*. McGraw-Hill New York, 3rd ed., 1976.

- [77] S. Osher and C.-W. Shu, *High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations*, *SIAM Journal on Numerical Analysis* **28** (1991), no. 4 907–922.
- [78] M. Fashoro, O. Hajek, and K. Loparo, *Controllability properties of constrained linear systems*, *Journal of Optimization Theory and Applications* **73** (1992), no. 2 329–346.
- [79] L. S. Pontryagin, *Mathematical Theory of Optimal Processes*. Routledge, 2018.
- [80] J. Nocedal and S. Wright, *Numerical Optimization*. Springer Science & Business Media, 2006.
- [81] M. R. Kirchner, E. Ball, J. Hoffer, and D. Gaublonne, *Reachability as a unifying framework for computing helicopter safe operating conditions and autonomous emergency landing*, *IFAC-PapersOnLine* **53** (2020), no. 2 9282–9287.
- [82] M. R. Kirchner, *A level set approach to online sensing and trajectory optimization with time delays*, in *IFAC-PapersOnline*, vol. 52, pp. 301–306, 2019.
- [83] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT press, 3rd ed., 2009.
- [84] M. Shamaiah, S. Banerjee, and H. Vikalo, *Greedy sensor selection: Leveraging submodularity*, in *49th IEEE Conference on Decision and Control (CDC)*, pp. 2572–2577, IEEE, 2010.
- [85] S. T. Jawaid and S. L. Smith, *Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems*, *Automatica* **61** (2015) 282–288.
- [86] A. Krause and C. Guestrin, *Near-optimal observation selection using submodular functions*, in *AAAI*, vol. 7, pp. 1650–1654, 2007.
- [87] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta, *Robust submodular observation selection*, *Journal of Machine Learning Research* **9** (2008), no. Dec 2761–2801.
- [88] S. Shankar, K. Ezal, and J. P. Hespanha, *Finite horizon maximum likelihood estimation for integrated navigation with RF beacon measurements*, *Asian Journal of Control* **21** (2019), no. 4 1470–1482.
- [89] F. Bian, D. Kempe, and R. Govindan, *Utility based sensor selection*, in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, pp. 11–18, ACM, 2006.
- [90] W. J. Welch, *Branch-and-bound search for experimental designs based on D-optimality and other criteria*, *Technometrics* **24** (1982), no. 1 41–48.

- [91] S. Joshi and S. Boyd, *Sensor selection via convex optimization*, *IEEE Transactions on Signal Processing* **57** (2008), no. 2 451–462.
- [92] R. D. Gill and B. Y. Levit, *Applications of the van Trees inequality: a Bayesian Cramér-Rao bound*, *Bernoulli* **1** (1995), no. 1/2 59–79.
- [93] T. H. Summers, F. L. Cortesi, and J. Lygeros, *Corrections to "On submodularity and controllability in complex dynamical networks"*, *IEEE Transactions on Control of Network Systems* **5** (2018), no. 3 1503–1503.
- [94] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, *An analysis of approximations for maximizing submodular set functions-I*, *Mathematical Programming* **14** (1978), no. 1 265–294.
- [95] L. Malagò and G. Pistone, *Information geometry of the Gaussian distribution in view of stochastic optimization*, in *Proceedings of the ACM Conference on Foundations of Genetic Algorithms XIII*, pp. 150–162, 2015.
- [96] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [97] E. A. Wan and R. Van Der Merwe, *The unscented Kalman filter for nonlinear estimation*, in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158, IEEE, 2000.
- [98] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, *Interacting multiple model methods in target tracking: A survey*, *IEEE Transactions on Aerospace and Electronic Systems* **34** (1998), no. 1 103–123.
- [99] M. R. Kirchner, K. Ryan, and N. Wright, *Maneuvering vehicle tracking with Bayesian changepoint detection*, in *2017 IEEE Aerospace Conference*, pp. 1–9, IEEE, 2017.
- [100] M. R. Kirchner, J. P. Hespanha, and D. Garagić, *Heterogeneous measurement selection for vehicle tracking using submodular optimization*, in *2020 IEEE Aerospace Conference*, pp. 1–10, IEEE, 2020.
- [101] S. Osher and M. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.
- [102] R. E. Bellman, *Adaptive control processes: A guided tour.*, Princeton University Press (1961).
- [103] D. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.

- [104] D. Onken, L. Nurbekyan, X. Li, S. W. Fung, S. Osher, and L. Ruthotto, *A neural network approach applied to multi-agent optimal control*, in *European Control Conference (ECC)*, pp. 1036–1041, IEEE, 2021.
- [105] S. Bansal and C. J. Tomlin, *Deepreach: A deep learning approach to high-dimensional reachability*, in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1817–1824, IEEE, 2021.
- [106] M. Chen, S. L. Herbert, M. S. Vashishtha, S. Bansal, and C. J. Tomlin, *Decomposition of reachable sets and tubes for a class of nonlinear systems*, *IEEE Transactions on Automatic Control* **63** (2018), no. 11 3675–3688.
- [107] M. R. Kirchner, M. J. DeBord, and J. P. Hespanha, *A Hamilton–Jacobi formulation for optimal coordination of heterogeneous multiple vehicle systems*, in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11623–11630, IEEE.
- [108] M. Shirazi and A. Vosoughi, *On Bayesian Fisher information maximization for distributed vector estimation*, *IEEE Transactions on Signal and Information Processing over Networks* **5** (2019), no. 4 628–645.
- [109] J. Yong and X. Y. Zhou, *Stochastic Controls: Hamiltonian Systems and HJB equations*, vol. 43. Springer Science & Business Media, 1999.
- [110] H. Federer, *Geometric Measure Theory*. Springer, 1996.
- [111] G.-S. Jiang and D. Peng, *Weighted ENO schemes for Hamilton–Jacobi equations*, *SIAM Journal on Scientific Computing* **21** (2000), no. 6 2126–2143.
- [112] P. D. Lax and R. D. Richtmyer, *Survey of the stability of linear finite difference equations*, *Communications on Pure and Applied Mathematics* **9** (1956), no. 2 267–293.
- [113] R. Courant, K. Friedrichs, and H. Lewy, *On the partial difference equations of mathematical physics*, *IBM Journal of Research and Development* **11** (1967), no. 2 215–234.
- [114] M. G. Crandall and P.-L. Lions, *Two approximations of solutions of Hamilton–Jacobi equations*, *Mathematics of Computation* **43** (1984), no. 167 1–19.
- [115] J. F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*. Springer Science & Business Media, 2000.
- [116] K. B. Petersen and M. S. Pedersen, *The matrix cookbook*, *Technical University of Denmark* **7** (2008), no. 15 510.

- [117] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, *CasADi – A software framework for nonlinear optimization and optimal control*, *Mathematical Programming Computation* **11** (2019), no. 1 1–36.