

Lawrence Berkeley National Laboratory

LBL Publications

Title

A Framework for Robust Assimilation of Potentially Malign Third-Party Data, and Its Statistical Meaning

Permalink

<https://escholarship.org/uc/item/2kn5x6w2>

Journal

IEEE Intelligent Transportation Systems Magazine, 12(3)

ISSN

1939-1390

Authors

Wright, Matthew A
Horowitz, Roberto

Publication Date

2020

DOI

10.1109/mits.2020.2994098

Peer reviewed

A Framework for Robust Assimilation of Potentially Malign Third-Party Data, and its Statistical Meaning

Matthew A. Wright and Roberto Horowitz

Abstract—This paper presents a model-based method for fusing data from multiple sensors with a hypothesis-test-based component for rejecting potentially faulty or otherwise malign data. Our framework is based on an extension of the classic particle filter algorithm for real-time state estimation of uncertain systems with nonlinear dynamics with partial and noisy observations. This extension, based on classical statistical theories, utilizes statistical tests against the system’s observation model. We discuss the application of the two major statistical testing frameworks, Fisherian significance testing and Neyman-Pearsonian hypothesis testing, to the Monte Carlo and sensor fusion settings. The Monte Carlo Neyman-Pearson test we develop is useful when one has a reliable model of faulty data, while the Fisher one is applicable when one may not have a model of faults, which may occur when dealing with third-party data, like GNSS data of transportation system users. These statistical tests can be combined with a particle filter to obtain a Monte Carlo state estimation scheme that is robust to faulty or outlier data. We present a synthetic freeway traffic state estimation problem where the filters are able to reject simulated faulty GNSS measurements. The fault-model-free Fisher filter, while underperforming the Neyman-Pearson one when the latter has an accurate fault model, outperforms it when the assumed fault model is incorrect.

I. INTRODUCTION

Intelligent transportation systems (ITS) have long relied on the use of real-time data to enable reactive and proactive operations and control. The widespread and growing use of real-time data, however, brings to ITS a problem that affects many domains in information sciences and engineering: these systems and methods can be fragile when their data are incorrect, either due to faults in the sensors or a feeding-in of malicious data by a hostile attacker (“spoofing”).

ITS researchers have shown that existing real-time control schemes are sensitive to errors in data. Some recent research even shows that faulty data can lead to actively harmful control. These vulnerabilities exist at both the small-scale, individual-vehicle level, and the multi-vehicle, infrastructural coordinative level. At the smaller scale, for example, [2] recently demonstrated the capability to drive a ship off-course via spoofed global navigation satellite system (GNSS) signals, evading detection by both the crew and a statistical spoofing detector. At the broader, infrastructural level, [24] showed how common road traffic control systems and algorithms (e.g., ramp meters and the programs that control the metering rate in response to observed traffic volumes) can be manipulated into causing complex and costly congestion patterns by taking control of their input data.

In this paper, we focus more on the larger-scale end of this spectrum. Types of ITS applications at this scale include the above-mentioned road traffic control systems, as well as fleet management and tracking systems in industry. Public and private management entities have both been quick to adopt the use of data from GNSS due to their ubiquitous availability and – especially for public bodies that wish to avoid the need for expensive installation and maintenance of sensing infrastructure – relatively low cost [16]. Many authors in the ITS community have investigated the use of vehicle-carried GNSS transponders for real-time road traffic observation and control [8], [18], [26], [27].

The work described in this paper was originally inspired by technical problems we encountered in our prior work in this area. In [28], [29], we report on our efforts to use anonymized *third-party* data from connected vehicles to estimate the state of traffic on a freeway. That is, the assimilation of records consisting of times, positions, and speeds from transponders near the freeway, but without certainty of the correctness of the data. For example, upon manual inspection, several records showed transponders with near-zero speeds in times and spaces we believed were not in congestion (e.g., possibly a stopped car), unrealistically fast movement, or speeds that better matched the congestion patterns on the freeway’s opposite direction. Using a standard particle filter [12] for state estimation, when some data are of very low probability, led to divergence of the state estimate from the true state, and in some extreme cases, numerical errors caused by floating-point underflow. In those situations, we want to be able to reject these data that would not improve our state estimate, in a principled manner.

We also sought to develop a method that could reject these malign measurements without having models for all types of faulty data. This paper describes two modifications to a familiar estimation algorithm, one applicable to the situation where a model of faults exists; and one where such a model does not exist, and the engineer only has a model for sensors’ correct behavior. These two modifications are based on two different mathematical theories of hypothesis testing.

In the broader picture, we argue that robustness to faulty data is essential to ITS schemes that make use of GNSS. ITS schemes often make use of GNSS position, velocity, and time (PVT) measurements, which are susceptible to many sources of error. These error sources include, to name a few, multipath propagation, non-line-of-sight tracking, signal blockage, tropospheric and ionospheric conditions, and a multiplicity of navigation filter implementations [13]. In other words, the presence of noise or faults in GNSS data for ITS could be considered the norm, rather than the exception, and

The authors are with the Department of Mechanical Engineering, the California Partners for Advanced Transportation Technology (PATH) Program, and Berkeley DeepDrive (BDD), University of California, Berkeley, CA, 94720, USA (email: {mwright,horowitz}@berkeley.edu)

system robustness to both modeled and unmodeled faults is desirable.

The rest of this paper is organized as follows. Section II introduces the framework of the *filtering* problem that forms the base for many studies of real-time transportation system estimation [8], [26], and reviews the popular *particle filter* algorithm that forms the base of our robustified estimation procedure. Section III briefly reviews the theoretical and historical background of hypothesis testing (which forms the core of our robustification), and introduces the two most common frameworks: *Fisherian* and *Neyman-Pearsonian*. Section IV goes into the mathematical details of the two frameworks, and describes modifications necessary to apply them to a Monte Carlo scheme like the particle filter. Section V merges the standard particle filter with our testing frameworks developed in Section IV. Section VI recalls our motivating problem of freeway traffic state estimation using third-party data, and presents some simulation results of the two testing-robustified particle filters on this difficult nonlinear estimation problem. Section VII concludes with some discussion on what we feel is this method's interesting fusion of data and model.

II. BACKGROUND OF THE FILTERING PROBLEM

A. State Estimation of Dynamic Systems

We use notation common to nonlinear discrete-time stochastic dynamic systems. Suppose we have some stateful system whose state evolves in time. Let $x_k \in R^N$ denote the state vector of the system at time k . The system state is not fully observed; instead what is observed at time k is a measurement vector $y_k \in R^{M_k}$ (the dimensionality having a subscript k implies we may obtain varying numbers of measurements at different times k). The state and observation vectors' temporal behavior are governed by stochastic update and output equations,

$$\begin{aligned} x_k &= \mathcal{F}_\theta(x_{k-1}) \\ y_k &= \mathcal{G}_\theta(x_k) \end{aligned} \quad (1)$$

with θ a parameter vector describing the randomness or process/measurement noise of \mathcal{F} and \mathcal{G} . An equivalent probabilistic notation may rewrite (1) as

$$X_k | (X_{k-1} = x_{k-1}) \sim f_\theta(x_k | x_{k-1}) \quad (2a)$$

$$Y_k | (X_k = x_k) \sim g_\theta(y_k | x_k) \quad (2b)$$

where, following conventions of probability, a capital letter (e.g., X_k) denotes a random vector and a lower-case letter (e.g., x_k) denotes the value of a particular realization. The functions on the RHS's of (2) are probability density functions (PDFs). More precisely, $f_\theta(\cdot)$ and $g_\theta(\cdot)$ are the PDFs of the conditional distributions for the random variables X_k given X_{k-1} and Y_k given X_k , respectively.

The model-based *filtering problem*, a classic problem in stochastic systems, is the problem of estimating the unknown system state $X_k(\forall k)$ from the known observation vectors Y_k [6]. This is often done iteratively forward in time, repeating a two-step process at each successive time k .

The first step is called the **prediction step**. Assuming that we have an estimate of the PDF of the random variable

$X_{k-1}|Y_{0:k-1}$ from the previous timestep, where $Y_{0:k-1}$ is shorthand for the set $\{Y_0, Y_1, \dots, Y_{k-2}, Y_{k-1}\}$, we can use (2a) to obtain

$$p_\theta(x_k | y_{0:k-1}) = \int f_\theta(x_k | x_{k-1}) p_\theta(x_{k-1} | y_{0:k-1}) dx_{k-1}. \quad (3)$$

The second step is called the **filtering step** or **update step**. Here, we use the obtained measurements y_k and (2b) to compute

$$p_\theta(x_k | y_{0:k}) = \frac{p_\theta(x_k | y_{0:k-1}) g_\theta(y_k | x_k)}{p_\theta(y_k | y_{0:k-1})} \quad (4)$$

where

$$p_\theta(y_k | y_{0:k-1}) = \int p_\theta(x_k | y_{0:k-1}) g_\theta(y_k | x_k) dx_k. \quad (5)$$

Note that (4) is a particular statement of Bayes' rule, with $p_\theta(x_k | y_{0:k-1})$, $g_\theta(y_k | x_k)$, $p_\theta(y_k | y_{0:k-1})$, and $p_\theta(x_k | y_{0:k})$ playing the role of the prior, likelihood, marginal likelihood, and posterior PDFs, respectively. Because of this, the iterative predict-update approach to filtering is sometimes called *recursive Bayesian estimation* [12].

For some simple classes of systems $f_\theta(\cdot)$, the computations in (3)-(5) are computable in closed form (the most well known example being that if both $f_\theta(\cdot)$ and $g_\theta(\cdot)$ are affine in the state x_k with additive white Gaussian noise, all PDFs in the recursion (3)-(5) can be computed exactly through simple matrix algebra, and is known as the Kalman Filter [14]). In more general settings with more complex system and noise behaviors, some numerical approximation is required.

B. Particle Filter

One popular approximation method when the integrals in (3) and (5) are difficult or computationally intractable is the *particle filter* [1], [6], [12]. A particle filter may be used even when there is no closed-form expression for $f_\theta(\cdot)$ (precluding many classic numerical integration schemes), but the PDF may be sampled repeatedly, such as by running a stochastic simulation many times.

A particle filter is constructed by replacing the PDFs for X_k in the filtering equations (3)-(5) with approximate PDFs, which we will denote with a hat (e.g., $\hat{p}_\theta(\cdot)$ for $p_\theta(\cdot)$). These approximate PDFs are made up of many discrete samples (also called particles) from the continuous PDF. The particles are generated by repeatedly sampling from $f_\theta(\cdot)$.

In other words, a particle filter can approximate continuous PDFs via discrete probability mass functions (PMFs). For example, a particle filter approximation of the posterior PDF $p_\theta(x_k | y_{0:k})$ (4) may be written

$$p_\theta(x_k | y_{0:k}) \approx \hat{p}_\theta(x_k | y_{0:k}) = \sum_{p=1}^P p_\theta(x_k^p | y_{0:k}) \delta_{x_k^p}(x_k) \quad (6)$$

where $p \in \{1, \dots, P\}$ denotes individual particles, or atoms of the discrete PMF, and $\delta_{x_k^p}(x_k)$ denotes a Dirac delta that places a unit mass on the point x_k^p (we use the subscript as a notational shorthand for the usual notation, $\delta_{x_k^p}(x_k) \triangleq \delta(x_k - x_k^p)$, where x_k denotes the argument to the "function" $\delta(\cdot)$ and x_k^p is the offset that moves the unit mass from $x_k = 0$).

Reviewing the two items in the summand of (6), we see that individual particles have an atom of probability mass placed in the state space of the system, x_k^p (where the superscript p denotes the p th particle), and an associated probability $p_\theta(x_k^p|y_{0:k})$. Summing up these particles results in a PMF with P discrete points, each with an associated probability.

Much like in the theoretical, closed-form version of recursive filtering (3)-(5), the particle filter proceeds in an iterative predict-then-update manner. As before, to estimate the system state at timestep k , we assume that we start with an approximate PDF from the previous timestep, $\hat{p}_\theta(x_{k-1}|y_{0:k-1})$ (note the hat indicating it is an approximation). This approximation has P individual particles. We can obtain a particle filter estimate of the prior PDF, $\hat{p}_\theta(x_k|y_{0:k-1})$, by plugging each particle's state value x_{k-1}^p into the stochastic system equation $\mathcal{F}_\theta(x_{k-1})$ (1) [12],

$$x_k^p = \mathcal{F}_\theta(x_{k-1}^p)$$

where the randomness of $\mathcal{F}_\theta(\cdot)$ means that

$$\mathcal{F}_\theta(x_{k-1}^p) \sim f_\theta(x_k|x_{k-1}^p).$$

Then, a particle filter approximation for the prior PDF is

$$\begin{aligned} p_\theta(x_k|y_{0:k-1}) &= \int f_\theta(x_k|x_{k-1})p_\theta(x_{k-1}|y_{0:k-1})dx_{k-1} \\ &\approx \sum_{p=1}^P p_\theta(x_{k-1}^p|y_{0:k-1})\delta_{\mathcal{F}_\theta(x_{k-1}^p)}(x_k) \\ &= \sum_{p=1}^P p_\theta(x_{k-1}^p|y_{0:k-1})\delta_{x_k^p}(x_k) \\ &= \hat{p}_\theta(x_k|y_{0:k-1}) \end{aligned} \quad (7)$$

and the particle filter approximation for the posterior PDF is found by plugging (7) into (4),

$$\begin{aligned} p_\theta(x_k|y_{0:k}) &= \frac{p_\theta(x_k|y_{0:k-1})g_\theta(y_k|x_k)}{p_\theta(y_k|y_{0:k-1})} \\ &\approx \frac{\hat{p}_\theta(x_k|y_{0:k-1})g_\theta(y_k|x_k)}{p_\theta(y_k|y_{0:k-1})} \\ &= \frac{\sum_{p=1}^P p_\theta(x_{k-1}^p|y_{0:k-1})\delta_{x_k^p}(x_k)g_\theta(y_k|x_k^p)}{p_\theta(y_k|y_{0:k-1})} \\ &= \frac{\sum_{p=1}^P p_\theta(x_{k-1}^p, y_k|y_{0:k-1})\delta_{x_k^p}(x_k)}{p_\theta(y_k|y_{0:k-1})} \\ &= \hat{p}_\theta(x_k|y_{0:k}). \end{aligned} \quad (8)$$

This posterior approximate PDF is thus made up of the same collection of Dirac deltas as the prior approximate PDF, $\hat{p}_\theta(x_k|y_{0:k-1})$, but with updated weights to reflect each point's posterior probability, after assimilating the measurement y_k through the likelihood.

As has been mentioned, the use of the particle filter avoids having to explicitly calculate difficult integrals. Of particular relevance is the calculation of the marginal likelihood $p_\theta(y_k|y_{0:k-1})$. Instead of using (5), we use the fact that in a PMF, the probabilities of all points must sum to one, to normalize the un-normalized probabilities $p_\theta(x_k^p|y_{0:k})$ in (8),

$$p_\theta(y_k|y_{0:k-1}) \approx \sum_{p=1}^P p_\theta(x_k^p|y_{0:k}). \quad (9)$$

In implementations of a particle filter, (8)-(9) make up the filtering step that is used in practice. However, as of yet, we have not brought into consideration the problem of measurement fault detection. When we introduce the framework for incorporating hypothesis tests for measurement fault detection in Section V, we will use a different update computation, one that includes an additional hypothesis-testing step.

As an important side note, we have omitted discussion of the particle filter's post-update resampling step because it is not immediately relevant here. See, e.g., [6], for details.

III. STATISTICAL TESTING: AN INTRODUCTION

Most readers of scientific literature are familiar with hypothesis testing in the form of reports of “p-values” and “statistical significance” in the context of discussions of, e.g., medical research. The most popular form of hypothesis test is the so-called “null hypothesis significance test” (NHST) [25]. In a NHST, a null hypothesis of, loosely speaking, “no relation” or “no correlation” is proposed. Then, a p-value for the data under this null hypothesis is computed, and if it is less than a hard boundary of, e.g., 5%, the test is said to have shown “statistical significance,” and a specified alternative hypothesis is accepted. The NHST is actually a fusion of two distinct theories [25]: *significance testing*, due to Fisher [9]–[11], and *hypothesis testing*, due to Neyman and Pearson [21]–[23].

It should be noted that concepts that are rooted in one of the two statistical testing frameworks, but do not make sense in the other, are often discussed alongside each other in the NHST presentation. For example, the Fisher framework only considers one hypothesis, the null hypothesis. On the other hand, in the Neyman-Pearson model, multiple hypotheses exist, along with Type I and Type II (also called false positive and false negative, respectively) error rates and statistical power, but p-values are absent (p-values are explicitly defined only in the Fisherian framework) [25].

The implications of this dichotomy are more than just philosophical and terminological: for some problems, strict adherence to one theory will lead to a different statistical test than would be derived using the other (see [17] for more discussion on the degree of these differences).

IV. STATISTICAL TESTING FOR MEASUREMENT REJECTION

A. Notation

For this section, where we review classical tests for measurement rejection and introduce new Monte-Carlo-based tests, we will use a somewhat simpler notation.

Suppose that we have data $D = \{d_1, \dots, d_n\}$, which come from a distribution with PDF $p_\theta(D)$. We use D instead of the classical X for data to avoid confusion with our system state variable. The PDF has an unknown parameter (or set of parameters) θ . The testing problem is to evaluate the likelihood of our data for certain values of θ and make decisions about whether those θ values should be used or not.

The remainder of this Section deals with the mathematical details of both the Fisherian and Neyman-Pearsonian theories described above. We will begin with the Neyman-Pearson

framework as its basic elements are likely more familiar to a reader with an applied knowledge of statistics.

B. Neyman-Pearsonian “hypothesis testing”

In this framework, in addition to our data D and PDF $p_\theta(D)$, we have two competing hypotheses: $H_0 : \theta = \theta_0$ and $H_1 : \theta = \theta_1$. In this case, where both hypotheses fully specify the form of the likelihood $p_\theta(D)$ (since each hypothesis consists of only a single point for θ), a ratio of the two hypotheses’ likelihoods might take the form

$$\Delta(D) = \frac{p_{\theta_1}(D)}{p_{\theta_0}(D)}. \quad (10)$$

A hypothesis test in this case is often called a “simple-vs-simple” hypothesis test (a simple hypothesis is one that fully specifies the model parameters). For the remainder of this discussion, we will focus on the simple-vs-simple tests. The formal extension to compound hypotheses is a part of future work.

A well-known result called the Neyman-Pearson Lemma [22] states that, for a given simple-vs-simple hypothesis testing problem, the optimal test (in that it minimizes the Type II error rate among all tests with a given Type I error rate¹) is a *likelihood ratio test*. A likelihood ratio test is one where the likelihood ratio $\Delta(D)$ is the test statistic of interest. A likelihood ratio test using the likelihood ratio given in (10) has the form

$$\psi(D) = \begin{cases} 1 & \text{if } \Delta(D) < c \\ 0 & \text{if } \Delta(D) > c \end{cases} \quad (11)$$

for some constant c . The test prescribes that, of the two hypotheses, we accept $H_{\psi(D)}$.

The constant c in (11) is chosen to set the likelihood ratio test to have a certain Type I error rate. This selected Type I error rate is called the test’s *significance level* and usually given the symbol α .

When the integral is computable, the constant c just mentioned is found by solving for it as a function of α in

$$\begin{aligned} E_{\theta_0} \psi(D) &= \int \psi(D) p_{\theta_0}(D) dD \\ &= P_{\theta_0}(\Delta(D) < c) = \alpha \end{aligned} \quad (12)$$

where we use the fact that the expectation of an indicator function (e.g., $\psi(D)$ (11)) is equal to the integral of the PDF p_{θ_0} over the set where the indicator function equals one.

Of critical importance in (12) is that the PDF of integration for $\psi(D)$ is the likelihood under H_0 . This is because the Type I error rate is defined as a rejection of H_0 when H_0 is *actually true*. This choice is made because, conventionally, H_0 represents the status quo, or a prior belief about θ before any evidence, and practitioners are interested in tests that have a small error probability when H_0 is correct [15].

Once we have selected our α (and therefore our c), we collect the data D , evaluate our likelihood ratio (10), and select either H_0 or H_1 depending on the value of $\psi(D)$ (11).

¹The Type I (“false positive”) error rate, $P(\text{Reject } H_0 | H_0 \text{ true})$, is the mathematical probability that H_0 is rejected, conditioned on it being true; and the Type II (“false negative”) error rate, $P(\text{Accept } H_0 | H_0 \text{ false})$, is the probability that H_0 is accepted, conditioned on it being false.

C. Monte-Carlo Neyman-Pearsonian Likelihood Ratio Tests

We now turn to how we must modify the Neyman-Pearson likelihood ratio test framework for use in our Monte Carlo framework. Suppose that rather than a known likelihood $p_\theta(D)$, we only have a set of samples $d_i \sim p_\theta(D)$, $i \in \{1, \dots, n\}$, as well as their associated likelihoods under the null hypothesis $p_{\theta_0}(d_i)$. Then, while we cannot solve the integral in (12) in closed form, we can still approximate it via Monte Carlo simulation,

$$\begin{aligned} E_{\theta_0} \psi(D) &= \int \psi(D) p_{\theta_0}(D) dD \\ &\approx \sum_{i=1}^n \psi(d_i) p_{\theta_0}(d_i) \\ &= \hat{E}_{\theta_0} \psi(D). \end{aligned} \quad (13)$$

Designing our test for a specific significance level (i.e., choosing c) in this case does not make sense, due to a lack of a closed-form integral equation in which to solve for c as a function of α . Instead, we propose to select either H_0 or H_1 based on the observed likelihood ratio statistic directly. The general idea is as follows. Considering (12) under the classic Neyman-Pearson framework, we would select c so that an α fraction of the probability mass distributed by $\hat{p}_{\theta_0}(D)$ returns values of D s.t. $\psi(D) = 1$ (this is what is shown in the second line of (12)). Under the empirical approximation (13), on the other hand, we have a finite amount of points, and can easily compute whether $\Delta(d_i)$ is below or above 1 for every i . The distribution of the probability mass in the empirical distribution is itself defined by our known sample weights $p_{\theta_0}(d_i)$. Therefore, we can determine whether at least an α portion of the probability mass under the empirical distribution recommends selecting H_1 by simply noting which samples have a higher likelihood under H_1 than under H_0 , and then adding up those samples’ H_0 -probabilities $p_{\theta_0}(d_i)$ and seeing whether this sum is greater or smaller than α .

Mathematically, the empirical Neyman-Pearson likelihood ratio test we propose is

$$\hat{\psi}(D) = \begin{cases} 1 & \text{if } \sum_{i=1}^n 1_{\{\Delta > 1\}}(d_i) p_{\theta_0}(d_i) < \alpha \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

where

$$1_{\{\Delta > 1\}}(d_i) = \begin{cases} 1 & \text{if } \Delta(d_i) > 1 \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

is an indicator function of whether d_i shows a higher likelihood under H_1 than under H_0 .

D. Fisherian “significance testing”

As mentioned above, the Fisherian formulation differs from the Neyman-Pearsonian one in several ways. One important difference is that it specifies the selection of only a single hypothesis, the null hypothesis H_0 , which specifies the PDF as $p_{\theta_0}(D)$. Usually, in a significance test, the null chosen is meant to be more interesting than the common “no relationship” hypothesis test, and reflects some *a priori* knowledge. Observing that the data D do not fit well with the null hypothesis

H_0 is meant to lead to reconsideration and indication to the practitioner that their prior assumptions used in crafting H_0 should be re-evaluated [25].

The Fisherian framework calls for the calculation of a test statistic of the data, $T(D)$. Unlike the simple-vs-simple case in the Neyman-Pearson framework (where we can apply the Neyman-Pearson Lemma), the optimal test statistic is not immediately given. Instead, its form depends on the particular form of the likelihood $p_{\theta_0}(D)$.

Here, we will assume that the null hypothesis H_0 is simple in the Neyman-Pearsonian sense, in that it fully specifies the likelihood: $H_0 : \theta = \theta_0$. Then, we want to compute the tail probability of the observed test statistic $T(D)$ under H_0 . This quantity is the p-value. Whether this tail probability will be a one-sided or two-sided value again depends on the particular form of the PDF $p_{\theta_0}(D)$. For the particular example of a two-sided test, the p-value (assuming $T(D) \in \mathbb{R}$) will be

$$\begin{aligned} \text{p-value} &= P_{\theta_0}(T < -|T(d)|) + P_{\theta_0}(T > |T(d)|) \\ &= \int_{-\infty}^{-|T(d)|} p_{\theta_0}(T(D)) dD + \int_{|T(d)|}^{\infty} p_{\theta_0}(T(D)) dD \end{aligned} \quad (16)$$

where $p_{\theta_0}(T(D))$ is the PDF of the statistic T (also called the statistic's sampling distribution) under H_0 and the lower-case formatting of d in $T(d)$ indicates that it is the actually-observed value of the random statistic $T(D)$.

For some common sampling distributions like the univariate Gaussian, Student's t , and χ^2 distributions, the solutions to the tail probability integrals in (16) are available in the familiar statistical testing reference tables, or can be readily computed via statistical software.

E. Monte-Carlo Fisherian Significance Tests

We can move from the theoretical framework of continuous integrals with closed-form solutions (16) to the Monte Carlo framework using similar arguments as in Section IV-C.

Considering (16), we see the same type of integral we had in (12) (recalling that the test term $\psi(D)$ (11) acted as an indicator function for a one-sided interval, effectively performing the same function as the one-sided limits of integration in (16)). Therefore, using similar arguments as before, we can get an approximation of the Fisherian p-value from (16) (noting the finite integration bound $T(d)$ is replaced with the approximation \hat{T}) as

$$\widehat{\text{p-value}} = \int_{-\infty}^{-|\hat{T}|} p_{\theta_0}(T(D)) dD + \int_{|\hat{T}|}^{\infty} p_{\theta_0}(T(D)) dD \quad (17)$$

where

$$\hat{T} = \sum_{i=1}^n T(d_i) p_{\theta_0}(d_i). \quad (18)$$

Like in the Monte Carlo Neyman-Pearson framework, we have again created a weighted-average statistic using our weighted-average approximation of the PDF for the data.

Fisher himself advocated against the use of fixed levels and hard accept/reject boundaries, but instead suggested reporting the p-values directly (for more on this contrast, see, e.g., the discussions in [17, Sec. 4], [25, p. 415]). However, for our

current purposes this is not easily implementable because in our filtering context, we are trying to make a decision as to whether to accept or reject a measurement as we receive it. Taking a "soft" view, and considering a range of state space values based on our current range of belief of whether we should accept or reject H_0 , while potentially giving us a view of a broader range of possibilities (and separate measurement hypotheses) that we could revisit in light of future data, leads to a blow-up when the number of separate sensors increase [29]. Instead, for expedience, in what follows we adopt the Neyman-Pearson and null hypothesis significance test and select a hard significance level α , and reject or accept the measurement based on whether our estimated p-value (17) is larger or smaller. A more "inductive" approach, closer to the spirit of the original Fisherian significance test, that updates H_0 based on repeated tests of measurements from the same sensor, is an avenue for future work.

V. A PROBABILISTIC OUTLIER-REJECTING PARTICLE FILTER

This section unifies the particle filtering framework reviewed in Section II and the hypothesis testing methods developed in Section IV. As mentioned in Section IV-A, we will unify these ideas in the notation of the state estimation problem. In this paper, we will not exhaustively define all PDFs of interest in the interest of readability. See [29] for a more lengthy discussion of a precursor to the Fisher-type hypothesis-testing particle filter discussed in the present paper.

Recall the filtering or update step in the particle filtering algorithm (8). In our prior discussion, we considered a measurement likelihood $g_{\theta}(y_k|x_k)$ (2b). In this notation, we are stating that the random measurement vector $Y_k|X_k$ has a joint distribution across all dimensions. This makes sense if the measurement noises of the different elements of the measurement vector are correlated or otherwise dependent.

Considering our problem of needing to assimilate data from multiple third-party sensors, though, it makes sense to assume a conditional independence of the measurements. If we say that at time k , we receive measurements from M sensors, with sensor j 's measurement being the random variable Y_k^j , then by assuming conditional independence of the sensors given X_k , we can write

$$Y_k|(X_k = x_k) \sim g_{\theta}(y_k|x_k) = \prod_{j=1}^M g_{\theta}^j(y_k^j|x_k). \quad (19)$$

This conditional independence assumption is a common assumption in multisensor filtering and sensor fusion (e.g., [3], [7], [19]). Examining (19), we notice that we have factored our particle filter likelihood (8) into per-sensor PDFs.

Considering a sensor likelihood $g_{\theta}^j(y_k^j|x_k)$, we can parameterize its non-fault, faulty (for one or more known types of fault, if applicable), spoofed, etc. behavior in θ . And, if we have models for one of these behaviors, we can accordingly form hypotheses: $H_0 : \theta = \theta_0$, $H_1 : \theta = \theta_1$, etc. This is how we bring together the particle-filtering and Monte Carlo fault-detection theories. Each individual particle, which has a value of the random variable $Y_k^j|X_k$ and a probability (the particle's

weighting in the collection of particles), serves as a sample (a d_i in Section IV’s terminology). Repurposing these particles as datapoints for the expected behavior of the data under the stated hypotheses lets us reject, in real time, measurements that do not match our prior models for data that would come from a correctly-functioning sensor.

Based on the above discussion, the general framework for the robustified particle filter is given below.

- 1) Perform a prediction step as normal, using (7).
- 2) For each sensor j at time k , calculate either the likelihood ratio or Fisher test statistic, depending on whether a Neyman-Pearson or Fisher test is used.
- 3) For each sensor, determine whether to reject it as faulty using the relevant hypothesis test for the actually-observed measurement and selected α .
- 4) Perform an update step with the non-rejected measurements using (8) (and, if desired, a resampling step).
- 5) Advance in time, $k \leftarrow k + 1$, return to step 1, repeat.

The type of test (Fisherian or Neyman-Pearsonian) to select for each situation and each sensor is dependent on the problem circumstances. We believe that it generally makes sense to favor a Neyman-Pearson-type test when one has trustworthy models for all reasonably-expected types of faults, and a Fisherian test when one does not. Our example case study presented in the next section shows some results for different types of tests and different fault models of varying accuracy.

VI. EXAMPLE APPLICATION

As mentioned in the introduction, our work presented above was motivated by earlier work involving the use of a particle filter and various data sources (some obviously faulty to a human when examined *post facto*) to estimate a freeway’s traffic state [28]. In this Section, we present a simulation case study based on that work, to demonstrate in particular the hypothesis-testing particle filters proposed in this paper.

A. Implementation details

Our system of study is a 19-mile portion of I-210 West in southern California. As our system model $f_\theta(\cdot)$, we make use of the macroscopic Cell Transmission Model (CTM) [4], which approximates traffic as compressible fluid flows. This type of model can capture important nonlinear emergent features in traffic flows like traffic jams and congestion waves.

In the CTM, the freeway is discretized into a sequence of finite-volume cells, also called links. The state vector x_k is the vector of link densities $\rho_{\ell,k}$, where ℓ indexes the links (with $\ell + 1$ immediately downstream of ℓ) and k is the time index. Link ℓ ’s state update equation is

$$\rho_{\ell,k+1} = \rho_{\ell,k} + \frac{1}{L_\ell} (q_{\ell-1,k} - q_{\ell,k} + r_{\ell,k} - s_{\ell,k}) \quad (20)$$

where L_ℓ is the length of link ℓ , $q_{\ell,k}$ denotes the vehicle flow going from link ℓ to link $\ell + 1$ at time k , $r_{\ell,k}$ is the flow entering link ℓ from an onramp (if any) at time k , and $s_{\ell,k}$ is the flow leaving link ℓ to an offramp (if any) at time k .

When there is no onramp or offramp between links ℓ and $\ell + 1$, the flow from link ℓ to link $\ell + 1$, $q_{\ell,k}$ in (20), is given by

$$q_{\ell,k} = \min(v_{f,\ell} \cdot \rho_{\ell,k} \cdot L_\ell, Q_{max,\ell}, w_{\ell+1} \cdot L_{\ell+1} \cdot (\rho_{J,\ell+1} - \rho_{\ell+1,k})), \quad (21)$$

where $v_{f,\ell}$ is the freeflow speed of link ℓ ; $Q_{max,\ell}$ is the capacity, or maximum possible flow over a time period, of link ℓ ; $w_{\ell+1}$ is the speed at which congestion waves propagate upstream in link $\ell + 1$; and $\rho_{J,\ell+1}$ is the jam density, or maximum possible density, of link $\ell + 1$. The third argument in the $\min(\cdot)$ function in (21) lets the downstream link $\ell + 1$ refuse to accept flow from link ℓ if $\ell + 1$ is too full.

When there is an onramp and/or an offramp between links ℓ and $\ell + 1$, we determine the flow $q_{\ell,k}$ according to the junction model of [20]. The ramp flows themselves, $s_{\ell,k}$ and $r_{\ell,k}$ in (20), are random variables. See [28] for full implementation details of these last two points.

A common type of first-party sensor for freeway traffic are inductive loop detectors buried in the pavement. These detectors can noisily measure density. A third-party source of data are vehicle-carried GNSS devices that report the speed of individual vehicles. In the CTM, the speed of traffic in link ℓ at time k is $v_{\ell,k} = L_\ell \cdot \rho_{\ell,k} / q_{\ell,k}$. A high vehicle density leads to congestion, and hence low speeds. We can use speed measurements to estimate density using this relationship in a Rao-Blackwellized particle filter [5].

To test our fault detection method, we simulated a realization of our freeway model, with randomness introduced by the random onramp, offramp, and upstream boundary flows. In addition to noisy density measurements from 41 loop detectors, we simulated GNSS speed measurements with a simulated penetration rate of 2% (i.e., each vehicle had a 2% chance of noisily reporting its speed). To generate the faulty third-party measurements, we gave each speed measurement a 30% probability of being faulty. We used two fault models: a faulty measurement had a 1/3 probability of reporting zero (i.e., a stopped car misreporting its location), and a 2/3 probability of drawing from a Gaussian distribution with mean 30 m/s and standard deviation 10 m/s. The non-fault model for velocity measurements, $g_{\theta_0}^j(\cdot)$, was Gaussian with a mean of the true link velocity and standard deviation of 20% of the mean (similar to [27]). Fig. 1 shows the true state and velocity measurements used.

B. Results

We tested both the Fisherian and two instances of the Neyman-Pearsonian (each using a different fault hypothesis H_1) against this problem. The results are presented in Tables I through III.

In the tables, we report both the performance of the statistical tests in rejecting faulty measurements, as well as the resulting estimation error obtained when using the non-rejected measurements. “Positives” refer to measurements for which we rejected H_0 , i.e., sensors that our fault decision statistical test concluded were faulty. “True” and “False” refer to correct and incorrect decisions, respectively, of whether a measurement

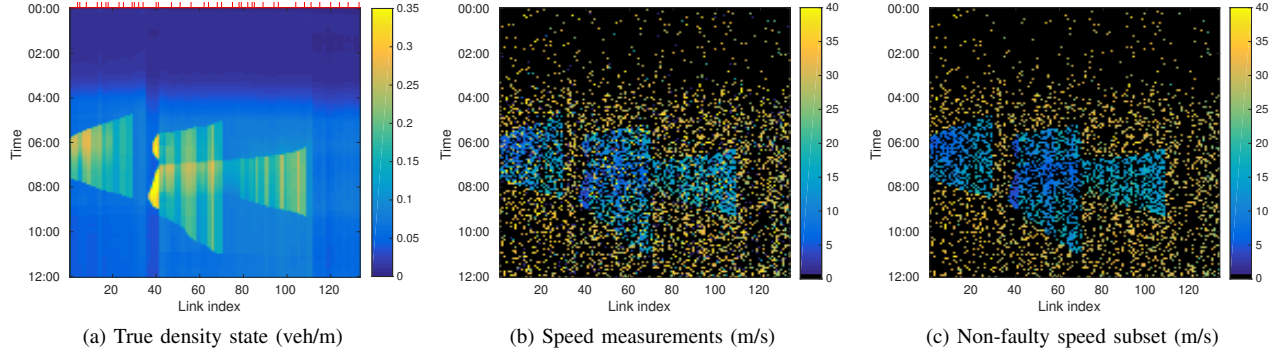


Fig. 1. Simulated true density state trajectory (a), speed measurements (b), and non-faulty subset of speed measurements (c) used in simulation. Traffic moves to the right, and the time period considered is midnight to noon (as marked on the vertical axis). In (a), the links instrumented with loop detectors that noisily measure density are marked with red ticks. At peak morning demand, bottlenecks near links 30, 70, and 110 lead to traffic jams that propagate upstream (i.e., they extend to the left as time advances), leading to increased density and lower speed. The jams later dissipate as demand falls.

is faulty. “Labeling error” reports the overall percentage of incorrect decisions (False Negatives and False Positives). The estimation error is reported in terms of the mean absolute percentage error (MAPE), the average of $|\hat{\rho}_{\ell,k} - \rho_{\ell,k}|/\rho_{\ell,k}$ for all ℓ and k , with $\hat{\rho}_{\ell,k}$ the ℓ th entry of $\hat{\rho}_k = \sum_{p=1}^P x_k^p \cdot p_{\theta}(x_k^p|y_k)$, i.e., the mean of the posterior particle filter PDF.

The two Neyman-Pearson estimation experiments are shown in Tables I and II. As indicated in the table names, Table I shows results for a simulation where the likelihood ratio test had an incorrect faulty measurement likelihood, and Table II one with the correct faulty measurement likelihood. The faulty measurement likelihood of Table I was a Gaussian that only placed mass near zero, i.e., it was crafted to select the fake “stopped car” vehicles. It indeed rejects the “stopped car” vehicles, but, as reflected in the relatively high labeling error rates, it does not reject the purely random measurements. On the other hand, the Neyman-Pearson fault detector with the correct H_1 model, unsurprisingly, performs better, rejecting many of the stopped-car and the purely random measurements. The Fisherian results (Table III) show that the estimation accuracy is quite sensitive to the selected α . There are much larger changes in labeling error and MAPE across the scale of α values selected than for either of the Neyman-Pearson results. This is not too surprising, as not having any alternative hypothesis H_1 to compare against, makes the p-value much more sensitive to small variations in the likelihood under H_0 than the likelihood ratio would be.

Of particular interest are the columns for $\alpha = 0.001$ and 0.01 in the Fisher results table (Table III). For these values of α , we obtain results that are between the performance of the correct- and incorrect-fault-model Neyman-Pearson filters. This is an encouraging result, as it confirms our intuition that for a properly tuned α , not having a fault model can beat the performance of using an incorrect one.

As mentioned in the caption for the tables, a particle filter estimator that did not see any faulty data (i.e., the true measurement distribution was $g_{\theta_0}^j(\cdot)$) obtained a MAPE of 3.43%. Unsurprisingly, none of the fault-detecting estimators consistently managed to obtain this level of accuracy, although the Neyman-Pearson fault detector with the correct fault model did come within a standard deviation or two.

TABLE I
NEYMAN-PEARSON FAULT DETECTION/ESTIMATION (INCORRECT H_1)

	$\alpha = 0.001$	$\alpha = 0.01$	$\alpha = 0.1$
True Positives	700 \pm 0	700 \pm 0	700.40 \pm 0.55
False Positives	58.40 \pm 9.50	62.80 \pm 5.72	76.60 \pm 7.02
True Negatives	4535.60 \pm 9.50	4531 \pm 5.72	1305.60 \pm 7.02
False Negatives	1306 \pm 0	1306 \pm 0	1305.60 \pm 0.55
Labeling Error (%)	20.67 \pm 0.14	20.74 \pm 0.09	20.94 \pm 0.10
Density MAPE (%)	3.80 \pm 0.12	3.86 \pm 0.05	3.93 \pm 0.11

TABLE II
NEYMAN-PEARSON FAULT DETECTION/ESTIMATION (CORRECT H_1)

	$\alpha = 0.001$	$\alpha = 0.01$	$\alpha = 0.1$
True Positives	1415.80 \pm 5.12	1433.80 \pm 1.79	1450.80 \pm 2.68
False Positives	94.80 \pm 8.07	106.20 \pm 16.80	118.20 \pm 17.61
True Negatives	4499.20 \pm 8.07	4487.80 \pm 16.80	4475.80 \pm 17.61
False Negatives	590.20 \pm 5.12	572.20 \pm 1.79	555.20 \pm 2.68
Labeling Error (%)	10.38 \pm 0.18	10.28 \pm 0.28	10.20 \pm 0.31
Density MAPE (%)	3.51 \pm 0.08	3.53 \pm 0.18	3.57 \pm 0.17

TABLE III
FISHER FAULT DETECTION/ESTIMATION

	$\alpha = 0.001$	$\alpha = 0.01$	$\alpha = 0.1$
True Positives	1214.80 \pm 2.49	1294.60 \pm 4.62	1457 \pm 3.32
False Positives	39 \pm 2.74	76.40 \pm 12.10	349.40 \pm 29.52
True Negatives	4555 \pm 2.74	4517.60 \pm 12.10	4244.60 \pm 29.52
False Negatives	791.20 \pm 2.49	711.40 \pm 4.62	549 \pm 3.32
Labeling Error (%)	12.58 \pm 0.05	11.94 \pm 0.24	13.61 \pm 0.49
Density MAPE (%)	3.66 \pm 0.10	3.71 \pm 0.18	4.22 \pm 0.21

“Positives” refer to sensors for which we rejected H_0 , i.e., sensors that our fault detection statistical test concluded were faulty. “True” and “False” refer to correct and incorrect decisions, respectively, of whether a sensor is faulty. In a simulation with no faulty velocity measurements, a lower-bound density MAPE of 3.43% was achieved.

All values reported are the mean and standard deviation of five identical simulations with different random seeds.

MAPE = mean absolute percentage error.

VII. CONCLUSION

This article presented a principled fault-detecting particle filter for real-time rejection of potentially faulty measurements. Our methods are based on the classical Fisherian and Neyman-Pearsonian statistical testing theories, and follows these theories to arrive at different testing methods for when the engineer has a reliable model of faults, and when she does not.

One item of interest is the subtle inversion of what is considered the “data” in our hypothesis tests. Notice that in this paper, the “data” that we use to estimate our test statistic are actually the simulated particles, which come from the system model, and we perform our hypothesis test to accept or reject the observations. An interesting implication of this nuance is how this hypothesis-testing particle filter allows a highly-trusted model to overpower the data, whereas a standard particle filter will always accept every datapoint, even if it is a clear outlier. We believe that the proposed techniques are closely aligned with the contemporary effort to fuse model-based and data-driven estimation and control techniques in many information sciences: to bring priors obtained from the engineering discipline to the surge of “big data.” These lessons are likely to be useful in many areas to bring GNSS and other “big data” to ITS and other built-environment applications.

ACKNOWLEDGEMENTS

This research was supported by the National Science Foundation under grant CNS-1545116 and Berkeley DeepDrive. M. A. W. thanks Adityanand Guntuboyina for raising a question regarding the congruity of [29]’s material with the Neyman-Pearson framework that helped direct some of the inquiry reported in the present paper. We also thank our colleague Alex A. Kurzhanskiy as well as three anonymous reviewers for their useful feedback.

REFERENCES

- [1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [2] J. Bhatti and T. E. Humphreys, “Hostile Control of Ships via False GPS Signals: Demonstration and Detection,” *Navigation*, vol. 64, no. 1, pp. 51–66, Mar. 2017.
- [3] J. F. Chamberland and V. V. Veeravalli, “Decentralized detection in sensor networks,” *IEEE Trans. Signal Process.*, vol. 51, no. 2, pp. 407–416, Feb. 2003.
- [4] C. F. Daganzo, “The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory,” *Transportation Research Part B: Methodological*, vol. 28, no. 4, pp. 269–287, Aug. 1994.
- [5] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, “ Rao-blackwellised Particle Filtering for Dynamic Bayesian Networks,” in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*. Stanford, California: Morgan Kaufmann Publishers Inc., 2000, pp. 176–183.
- [6] A. Doucet and A. M. Johansen, “A tutorial on particle filtering and smoothing: Fifteen years later,” in *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press, 2011, pp. 656–704.
- [7] H. Durrant-White and T. C. Henderson, “Multisensor Data Fusion,” in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2016, pp. 867–896.
- [8] A. Ferrara, S. Sacone, and S. Siri, *Freeway Traffic Modelling and Control*. Cham: Springer International Publishing, 2018.
- [9] R. A. Fisher, “The Logic of Inductive Inference,” *Journal of the Royal Statistical Society*, vol. 98, no. 1, pp. 39–82, 1935.
- [10] —, *Statistical Methods for Research Workers*. Oliver & Boyd, 1925.
- [11] —, *The Design of Experiments*. Oliver & Boyd, 1935.
- [12] N. Gordon, D. Salmond, and A. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proc. F Radar Signal Process. UK*, vol. 140, no. 2, p. 107, 1993.
- [13] S. Jin, E. Cardellach, and F. Xie, *GNSS Remote Sensing: Theory, Methods and Applications*. Dordrecht: Springer, 2014.
- [14] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.
- [15] R. Keener, *Theoretical Statistics: Topics for a Core Course*, 1st ed. New York, NY: Springer, 2010.

- [16] A. A. Kurzhanskiy and P. Varaiya, “Traffic management: An outlook,” *Economics of Transportation*, vol. 4, no. 3, pp. 135–146, Sep. 2015.
- [17] E. L. Lehmann, “The Fisher, Neyman-Pearson Theories of Testing Hypotheses: One Theory or Two?” *Journal of the American Statistical Association*, vol. 88, no. 424, pp. 1242–1249, Dec. 1993.
- [18] E. Lovisari, C. Canudas de Wit, and A. Y. Kibangou, “Density/Flow reconstruction via heterogeneous sources and Optimal Sensor Placement in road networks,” *Transportation Research Part C: Emerging Technologies*, vol. 69, pp. 451–476, Aug. 2016.
- [19] L. Mihaylova, R. Boel, and A. Hegyi, “Freeway traffic estimation within particle filtering framework,” *Automatica*, vol. 43, no. 2, pp. 290–300, Feb. 2007.
- [20] A. Muralidharan, G. Dervisoglu, and R. Horowitz, “Freeway traffic flow simulation using the Link Node Cell transmission model,” in *Proc. of the 2009 American Control Conference*, Jun. 2009, pp. 2916–2921.
- [21] J. Neyman and E. S. Pearson, “On the use and interpretation of certain test criteria for purposes of statistical inference: Part I,” *Biometrika*, vol. 20A, pp. 175–240, 1928.
- [22] —, “On the problem of the most efficient test of statistical hypotheses,” *Philosophical Transactions of the Royal Society of London A*, vol. 231, pp. 289–337, 1933.
- [23] —, “The testing of statistical hypotheses in relation to probabilities a priori,” *Proceedings of the Cambridge Philosophical Society*, vol. 29, pp. 492–510, 1933.
- [24] J. Reilly, S. Martin, M. Payer, and A. M. Bayen, “Creating complex congestion patterns via multi-objective optimal freeway traffic control with application to cyber-security,” *Transportation Research Part B: Methodological*, vol. 91, pp. 366–382, Sep. 2016.
- [25] J. W. Schneider, “Null hypothesis significance tests. A mix-up of two different theories: The basis for widespread confusion and numerous misinterpretations,” *Scientometrics*, vol. 102, no. 1, pp. 411–432, Jan. 2015.
- [26] T. Seo, A. M. Bayen, T. Kusakabe, and Y. Asakura, “Traffic state estimation on highway: A comprehensive survey,” *Annual Reviews in Control*, vol. 43, pp. 128–151, 2017.
- [27] D. B. Work, S. Blandin, O. P. Tossavainen, B. Piccoli, and A. M. Bayen, “A Traffic Model for Velocity Data Assimilation,” *Applied Mathematics Research eXpress*, vol. 2010, no. 1, pp. 1–35, Apr. 2010.
- [28] M. Wright and R. Horowitz, “Fusing Loop and GPS Probe Measurements to Estimate Freeway Density,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3577–3590, Dec. 2016.
- [29] M. A. Wright and R. Horowitz, “Particle-filter-enabled real-time sensor fault detection without a model of faults,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec. 2017, pp. 5757–5763.



Matthew A. Wright received the M.S. degree in Mechanical Engineering from the University of California, Berkeley in 2015, and is currently a candidate for the Ph.D. degree in Mechanical Engineering from the same institution.

His research interests include modeling, state estimation, and control for stochastic, complex, and networked systems such as vehicle traffic networks.



Roberto Horowitz received the Ph.D. degree in Mechanical Engineering in 1983 from the University of California, Berkeley. In 1982, he joined the Department of Mechanical Engineering, University of California, Berkeley, where he is currently the Department Chair and the James Fife Endowed Chair.

His research interests include the areas of adaptive, learning, nonlinear and optimal control, with applications to microelectromechanical systems (MEMS), computer disk file systems, robotics, mechatronics, and intelligent vehicle and highway systems.

Dr. Horowitz is the recipient of the 2018 ASME Rufus Oldenburger Medal.