

EXHIBIT 1019

Efficient Unbiased Tracking of Multiple Dynamic Obstacles Under Large Viewpoint Changes

Isaac Miller, *Member, IEEE*, Mark Campbell, *Member, IEEE*, and Daniel Huttenlocher

Abstract—A novel-tracking algorithm is presented as a computationally feasible, real-time solution to the joint estimation problem of data assignment and dynamic obstacle tracking from a potentially moving robotic platform. The algorithm implements a Rao–Blackwellized particle filter (RBPF) to factorize the joint estimation problem into 1) a data assignment problem solved via particle filter and 2) a multiple dynamic obstacle-tracking problem solved with efficient parametric filters. The parametric filters make use of a new target representation and stable features developed specifically for tracking full-size vehicles in a dense traffic environment. The algorithm is validated in real time, both in controlled experiments with full-size robotic vehicles and on data collected at the 2007 Defense Advanced Research Projects Agency (DARPA) Urban Challenge.

Index Terms—Detection and tracking of moving obstacles, field robots, Rao–Blackwellized particle filter (RBPF), sensor fusion.

I. INTRODUCTION

MODERN autonomous mobile robots operating at human scales of size face the significant challenge to perceive and understand the dynamic environment in which they (and we) interact. Fielded solutions to this perception problem must be both accurate and robust, as robotic failures in a human-populated operating environment can rapidly lead to disaster. The danger is particularly acute for full-size robotic vehicles, where tracking accidents and traffic accidents are separated only by a thin layer of actuators. Although better sensing hardware can improve perception of the current scene, autonomous systems benefit more from mitigating perception errors before they happen. The problem is, therefore, approached in the predictive language of data fusion, model-based estimation, and probabilistic analysis.

Historically, mobile robot dynamic obstacle-tracking techniques resemble the estimation techniques of classical target tracking. These approaches estimate the probability density

function of a target's state, i.e., a vector of random variables that evolve according to a model of the target's dynamics. The target's probability density is then occasionally updated with measurements taken from one or more noisy sensors. The framework extends naturally to track multiple targets, where the goal is to estimate the joint probability density across all target states. If the pairings between measurements and targets are known, the tracking problem is commonly solved with a traditional Kalman filter (KF) or one of its variants [1].

With unknown measurement/target pairings, both the measurement correspondences and the target states must be estimated simultaneously. This is an instance of the well-known data assignment problem, which is also called the correspondence problem or the data association problem [2]. Since the problem's exact solution is intractable, a vast array of approximations have been investigated. These approximations include simple nearest neighbor matching schemes (the maximum likelihood approach) [2], global nearest neighbor matching schemes (the joint compatibility test) [3], one- and multistep approximations when marginalizing out data assignments (the joint probabilistic data association filter and multiple hypothesis tracking schemes) [2], [4], [5], and even Monte Carlo approximations to either the full-joint target state density or just to the data assignments (particle filters, Markov chain Monte Carlo (MCMC) techniques, Rao–Blackwellized particle filters (RBPFs), and the probabilistic hypothesis density filter) [6]–[10].

Although elegant in their Bayesian formulation, these classical-tracking approaches assume the tracked targets to be point objects. This approximation is reasonable for long-range-tracking scenarios, such as air, sea, or space traffic monitoring, where the targets are significantly smaller than sensor error at the ranges of interest. The approximation breaks down in many robotics applications, however, where the targets are nearby obstacles or landmarks. The approximation is particularly bad for human-scale robotics applications, where the robot may need to maneuver within a meter or two of large obstacles, such as cars and buildings. Furthermore, typical sensors, such as laser rangefinders, radars, and optical cameras, have errors that are only a small fraction of the width of these obstacles.

The breakdown of the point-obstacle assumption yields two significant challenges not addressed in target-tracking literature. First, the data assignment problem becomes more difficult, because the observed shapes of obstacles change rapidly at close range. Second, the obstacles cannot be estimated as point objects. One common *ad hoc* approach uses a feature extraction algorithm to condense the environment into a set of point features to track. This is used in the simultaneous localization and mapping (SLAM) framework [11], [12], as well as in automotive

Manuscript received October 4, 2009; revised June 2, 2010; accepted October 3, 2010. Date of publication November 22, 2010; date of current version February 9, 2011. This paper was recommended for publication by Associate Editor J. Neira and Editor W. K. Chung upon evaluation of the reviewers' comments.

I. Miller is with Coherent Navigation, Inc., San Mateo, CA 94404 USA (e-mail: itm2@cornell.edu).

M. Campbell is with the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: mc288@cornell.edu).

D. Huttenlocher is with the Computer Science Department, Cornell University, Ithaca, NY 14853 USA (e-mail: dph@cs.cornell.edu).

Digital Object Identifier 10.1109/TRO.2010.2085490

tracking [13]–[21]. Another approach divides the environment into a grid that estimate the states of each grid cell as individual point targets. This has been applied to stochastic terrain mapping [22], [23] as well as to detection and tracking of moving obstacles (DATMO) [24]. A final approach operates on raw sensor data, using either old sensor measurements or the target probability density function to explain individual sensor measurements. This has been applied to scan-matching within the SLAM framework [25], [26] and to automotive tracking [27].

Feature extraction and tracking techniques are, by far, the most ubiquitous automotive techniques due to their desirable computational properties. Grid-based methods and scan-matching methods are significantly more expensive, because they must consider pairings between each raw sensor measurement and each tracked target. In contrast, feature extraction techniques reduce large quantities of raw data into a smaller set of organized metadata, which permits more expensive and sophisticated Bayesian treatments of the data assignment problem. Unfortunately, the benefits of feature extraction tradeoff against the irreversible, non-Bayesian operations performed on the raw sensor data. Since these operations are often inconsistent from frame to frame, the resultant target estimates suffer from strong estimation biases, inconsistencies, and even instabilities. Some of these errors have even resulted in catastrophic failures in fielded robotic systems [28].

The instability of existing feature-based target-tracking schemes in full-size robotics applications motivates the research presented herein. The tracking algorithm presented in the following sections has been developed to address the biasing problems in feature extraction and feature tracking without sacrificing the rigor of Bayesian solutions to the data assignment problem. Here, a real-time RBPF is combined with stable features and a novel target representation to track full-size vehicles from a moving robotic platform. Particular attention is paid to driving situations with large changes in perceived target shapes. This study makes four significant contributions. First, a novel target representation and stable features are developed to allow a formal, yet efficient, treatment of the joint data assignment and target-tracking problem. The resulting real-time algorithm fuses three sensing modalities common in automotive-tracking applications: laser rangefinders, millimeter-wave radars, and optical cameras. Second, and in contrast with existing feature-based-tracking algorithms, this study is statistically shown to produce accurate, unbiased target-state estimates on experimental field data with full-size vehicles. These experiments emphasize the common, yet challenging aspects of vehicle perception: when target vehicles occupy a significant portion of the sensor field of view and when the perceived shape of targets vehicles changes rapidly. Third, the proposed tracking algorithm is shown to make target likelihoods accurate enough to permit real-time implementation of a sophisticated Monte Carlo solution to the data assignment problem. Finally, the algorithm has been implemented in real-time and fielded on Cornell University's 2007 Defense Advanced Research Projects Agency (DARPA) Urban Challenge (DUC) robot, where it facilitated 60 miles of autonomous driving in a dynamic urban environment [29].

The remainder of the paper is organized as follows. Section II discusses the historical context of target tracking, mapping, and

Bayesian applications related to this paper. Section III discusses the details of the tracking algorithm, providing a theoretical foundation for the joint estimation problem of data assignment and obstacle tracking. Section IV presents experimental performance for the tracking algorithm, first in controlled full-size vehicle-tracking experiments and, then, in an uncontrolled environment full of traffic. Section V summarizes the performance of the tracking algorithm in the 2007 DUC, and Section VI concludes, the paper.

II. RELATED WORK

The algorithm presented in this paper couples computationally feasible Monte Carlo data assignment techniques with a novel target representation and stable features to address biasing effects of feature extraction in dynamic environments with full-size robots. Although the authors know of no existing work eliminating estimation artifacts related to non-Bayesian feature extraction techniques, there remains a large body of related work both in target tracking and in fielded full-size robotic systems. First, the 2007 DUC set the stage for a variety of related approaches for detection and tracking of full-size vehicles from robotic systems in an urban traffic setting. Aside from Cornell's robot, five others successfully finished the DUC. Each employed some aspects of dynamic obstacle detection. Carnegie Mellon's "Boss" utilizes a feature-tracking algorithm that uses known road geometry to identify partial vehicle measurements amidst raw laser rangefinder and radar data [30]. These measurements are then fused to create a best-fitting full-vehicle hypothesis; tracked vehicles are updated with the best-fitting hypothesis. In contrast, Stanford's "Junior" differences successive frames of laser rangefinder data to identify regions of space-containing moving objects. Each of these regions is then modeled as a rectangular car and tracked separately from the others with a particle filter [31]. Virginia Tech's "Odin" utilizes off-the-shelf feature-based tracking implemented in a set of two Ibeo ALASCA XT laser rangefinders and an Ibeo ALASCA A0 laser rangefinder [32]. The Ibeo laser rangefinders segment raw data into distinct objects, which are then fitted to rectangles in the ground plane. The parameters of these rectangles are then used to update the best-fitting tracked vehicle model [33]. The Massachusetts Institute of Technology's "Talos" applies a spatial-clustering algorithm to laser rangefinder data; a heuristic voting scheme is then used to associate clusters from frame to frame via a best-fit criterion. Velocity measurements are calculated from apparent motion in matched clusters' bounding boxes and then used to update cluster velocity estimates [34]. The University of Pennsylvania's "Little Ben" also segments laser rangefinder data but extracts local line features. These features are then tracked from frame to frame using a multihypothesis KF, with hypothesis testing performed to reduce the persistence of inconsistent vehicle hypotheses [35]. Although each approach provided sufficient tracking for a successful completion of the DUC, each was different from Cornell's approach. Most of these algorithms utilized best-fit or heuristic solutions to the data assignment problem, utilized only laser rangefinders, and depended on *ad hoc*, non-Bayesian feature extraction techniques.

Aside from the contributions made in the DUC, a number of works have studied the data assignment problem directly in the

Ex.-1019 - ADB Safegate et al.

ADB Safegate et al. v. Honeywell International Inc. – PGR2023-00053

Page 3

context of applications. Within the SLAM framework, for example, variants of nearest neighbor matching have been explored, including maximum likelihood approaches, hypothesis testing to provide measurement validation gates, and efficient global maximum likelihood techniques to match observed measurements with mapped landmarks [3], [11], [12], [36]. Reversible data assignment has also been developed for robotic mapping to allow map estimators several iterations to determine correct data assignments before marginalizing out past state estimates [37]. The data assignment problem has also appeared in computer vision literature, where, for example, MCMC techniques have been combined with the expectation maximization (EM) algorithm to solve frame-to-frame correspondences between features in structure from motion problems [38]. Efficient approximations for data assignment in the context of tracking for real-time surveillance purposes have also been studied [39], [40]. Each of these approaches focus on the data assignment problem directly and, in doing so, are peripherally related to the work presented here. However, the argument made here is that efficient, practical tracking requires not only a rigorous solution to the data assignment problem but a stable target representation and set of features suitable to tracking individual targets as well. The algorithm that follows is thus motivated from this existing work in data assignment but extends the concept to accurate, real-time tracking in a practical and challenging environment.

III. EFFICIENT, UNBIASED-TRACKING ALGORITHM

Derivation of the tracking algorithm begins by casting the dynamic urban perception problem in a Bayesian framework. To this end, a set of variables O describing obstacle positions and motions are estimated to permit basic robotic navigation and obstacle avoidance. In this context, the cardinality of the obstacle variables O implicitly represents the number of obstacles in the world and may change over time. In addition to the obstacle variables O , a set of data assignment variables A are also estimated. These assignment variables record the history of sensor measurements generated by each obstacle, thereby dividing the user robot's sensor measurements into historical sequences associated with each individual obstacle. Neither the obstacle variables O nor the assignment variables A are known with certainty in the unstructured environment, and therefore, both must be estimated. The present algorithm, therefore, estimates the joint probability density over these two sets of variables, conditioned on all available sensor measurements

$$p(A_k, O_k | Z_k) \quad (1)$$

where Z_k are the set of observed sensor measurements, and k represents an integer time index. The use of capital letters A_k , O_k , and Z_k indicates a full-time history of these quantities, from the filter's inception at 0 to the present time k . The joint probability density in (1) may also be conditioned on other relevant information, such as ego robot motion estimates, where available. This other information is omitted for brevity, although ego robot motion estimates will be addressed in Section III-B for tracking targets in a robot-centric coordinate frame.

Although general-filtering methods exist to estimate joint probability densities over multiple variables, the case of perception of a dynamic urban environment on a mobile robot presents several difficulties. First, the probability density in (1) is hybrid, because the data assignment variables A are discrete, while the obstacle variables O are continuous. Second, the number of potential data assignment histories grows exponentially in time, number of measurements, and number of obstacles, preventing exact probabilistic reasoning even in modest environments. Finally, obstacles moving in a dynamic urban environment may be intelligent and capable of executing complex nonlinear maneuvers over time. These challenges in concert make most traditional estimation approaches unsuitable for estimating the joint density in (1).

To make the estimation problem tractable amidst these challenges, the joint density in (1) is first factorized

$$p(A_k, O_k | Z_k) = p(A_k | Z_k) \cdot p(O_k | Z_k, A_k). \quad (2)$$

This factorization is exact and is similar to one made by Montemerlo *et al.* to estimate the joint density between an ego robot's pose and the positions of static landmarks in a variant of the SLAM problem [41]. Intuitively, the factorization made in (2) is beneficial because it decouples the discrete data assignment problem $p(A_k | Z_k)$ from the continuous tracking problem $p(O_k | Z_k, A_k)$. Although the discrete data assignment problem is still too large to be solved with exact inference techniques, the continuous tracking problem conditioned on known data assignments may be solved with less-expensive parametric filters, such as the KF or sigma point filter (SPF) [1], [2], [42].

A. Discrete Data Assignment Problem

Despite the factorization made in (2), the number of discrete data assignment permutations still grows exponentially in time, number of obstacles, and number of measurements. It is, therefore, computationally infeasible to evaluate the corresponding data assignment density $p(A_k | Z_k)$ exactly; therefore, approximation techniques are used instead. For the present algorithm, the discrete data assignment density $p(A_k | Z_k)$ is approximated by a small number of randomly drawn samples, utilizing Monte Carlo likelihood-weighted sampling techniques [43], [44]. The goal of the factorization made in (2) is to make these nondeterministic data assignment choices "obvious," reducing sample size requirements through the use of intelligent and inexpensive parametric solutions to the continuous tracking problem. This avoids the significant expense associated with treating the entire joint probability density in (1) directly as a random finite set (RFS), which is, otherwise, a popular representation when data assignments are highly ambiguous [9], [10].

The hybrid particle filter used here, which estimates a factorized probability density by combining Monte Carlo sampling techniques with closed-form parametric filters, is known as a RBPF; see, for example, Doucet *et al.* [45]. Särkkä *et al.* have studied such an estimator in the context of target tracking from a single-fixed sensor, where it is shown in simulation to be robust against target confusion and other classically challenging data assignment problems [7]. The present algorithm extends

that RBPF framework to track the motion of dynamic obstacles in an urban environment relative to a moving sensing platform, i.e., a ground robot, where nearby obstacles occupy much of the field of view and the perceived shapes of these obstacles change rapidly. The approach offers a more rigorous Bayesian solution to the joint estimation problem than maximum likelihood data assignments, and the solution is made practical through the RBPF's computationally efficient factorization of data assignment and obstacle tracking.

In the proposed algorithm, as with other particle filters derived for Monte Carlo data assignment, each randomly drawn particle stores one possible data assignment history [44]. The particles are drawn according to a proposal density $q(A_k|Z_k)$, selected for its efficient sampling algorithms and its similarity to $p(A_k|Z_k)$. The particles drawn from this proposal density represent an approximation of the true density $p(A_k|Z_k)$ for the purposes of approximate inference

$$p(A_k|Z_k) \approx \sum_i w_k^i \delta(A - A_k^i) \quad (3)$$

where w_k^i is the likelihood weight of the i th particle A_k^i at time index k

$$w_k^i = \frac{p(A_k^i|Z_k)}{q(A_k^i|Z_k)} \quad (4)$$

and $\delta(\cdot)$ is the Kronecker delta function because data assignment is a discrete estimation problem. Note the weights w_k^i must sum to unity to preserve the density's normalization

$$\sum_i w_k^i = 1. \quad (5)$$

With (3) providing an approximate representation of $p(A_k|Z_k)$ and a closed-form parametric filter used to represent $p(O_k|Z_k, A_k)$, the full-joint probability density from (2) is written approximately as follows:

$$p(A_k, O_k|Z_k) \approx \sum_i w_k^i \delta(A - A_k^i) p(O_k|Z_k, A_k^i) \quad (6)$$

where the obstacle densities $p(O_k|Z_k, A_k^i)$ are conditioned on the specific data assignment history A_k^i of a particular particle. In essence, each particle in (6) represents a complete hypothesis about the ego robot's urban environment, including both a history of data assignment decisions as well as a parametric filter estimating states of obstacles, whose existence are implied by the particle's measurement assignments. Since each particle may have a different data assignment history, particles may have different estimates of obstacle states and may even have different numbers of obstacles.

Despite the complexity of each particle, the algorithm achieves computational feasibility by exploiting the success of parametric filters, such as the KF and SPF, for target tracking under known measurement assignments. This success allows data assignment to be decoupled from target tracking, and therefore, the algorithm only draws particles over discrete data assignments rather than the entire joint density $p(A_k, O_k|Z_k)$. Further simplifications to the dynamic urban perception problem can be made by realizing that the ego robot operating in real time only

requires an estimate of its surroundings at the present time. The problem is, therefore, simplified considerably by deleting measurement assignment and obstacle state histories from each particle after the information has been incorporated into the current estimate. This may be done by modifying the proposal density $q(A_k^i|Z_k)$, nominally a design choice "similar to" the true distribution $p(A_k|Z_k)$ to delete old information. In particular, if the proposal density $q(A_k|Z_k)$ is chosen to factorize as follows:

$$q(A_k|Z_k) = q(a_k|Z_k, A_{k-1}) q(A_{k-1}|Z_{k-1}) \quad (7)$$

then the likelihood weight from (4) can be expressed recursively [44]

$$w_k^i \propto \frac{p(z_k|Z_{k-1}, A_k^i) p(a_k^i|Z_{k-1}, A_{k-1}^i)}{q(a_k^i|Z_k, A_{k-1}^i)} w_{k-1}^i \quad (8)$$

where lowercase variables z_k and a_k indicate measurements and data assignments at a particular time index k . Equation (8) has been obtained, up to a normalization constant, by applying Bayes's rule to the data assignment density $p(A_k|Z_k)$. Note that both terms in the numerator of (8) are familiar quantities: $p(z_k|Z_{k-1}, A_k^i)$ is the likelihood of the obstacle-tracking filter after the k th data assignment decision in the i th particle, and $p(a_k^i|Z_{k-1}, A_{k-1}^i)$ is the predicted likelihood of the k th data assignment decision in the i th particle, before the k th measurement has actually been made.

One final step remains in deriving the particle filter for data assignment, and that is choosing the proposal density $q(a_k^i|Z_k, A_{k-1}^i)$ from which the random data assignment hypotheses for each particle are drawn. Because the density defines how the particles are sampled, it should be a density that can be sampled inexpensively. In particle-filtering applications that estimate the state of a dynamic system, the proposal density is often chosen based only on the system's transition model and its process noise, thereby ignoring the measurement history [44]. This choice is suboptimal in the sense that it does not minimize the sample variance on the particles' weights w_k^i , which more rapidly drives all but a few particle weights to zero under repeated renormalizations. In contrast, the optimal proposal density $q_{\text{opt}}(\cdot)$ minimizing the sample variance on the particles' weights has been shown to be the true density [44]. For the present data assignment problem, this density is as follows:

$$q_{\text{opt}}(a_k^i|Z_k, A_{k-1}^i) = p(a_k^i|Z_k, A_{k-1}^i). \quad (9)$$

In many cases, it is prohibitively expensive to sample from the optimal proposal density directly; therefore, a suboptimal proposal density is used instead [44]. In the special case of data assignment with parametric-tracking filters, however, the optimal proposal density may be sampled directly and inexpensively [7]. To derive the corresponding sampling algorithm, the optimal density is first rewritten using Bayes's rule

$$\begin{aligned} q_{\text{opt}}(a_k^i|Z_k, A_{k-1}^i) &= p(a_k^i|Z_k, A_{k-1}^i) \\ &= \alpha_k^i p(z_k|a_k^i, Z_{k-1}, A_{k-1}^i) \\ &\quad \times p(a_k^i|Z_{k-1}, A_{k-1}^i) \end{aligned} \quad (10)$$

where α_k^i is a normalizing constant that depends on the i th particle prior to sampling. The first of the remaining two terms $p(z_k | a_k^i, Z_{k-1}, A_{k-1}^i)$ is the likelihood of the measurement z_k , assuming it originated from a particular tracked obstacle in the i th particle. This term is the likelihood of the tracked obstacle filter at time index k , which is often accurately assumed to be Gaussian in parametric filters [1]. The second term $p(a_k^i | Z_{k-1}, A_{k-1}^i)$ is a transition model for the data assignments. This term represents any *a priori* assignment information that may be present before the measurement at time index k is actually made. In the present tracking implementation, this term is assigned a uniform probability across all data assignments. This choice is made to represent the fact that sensors are fused asynchronously, and the measurements, in general, have no predictable ordering. Note, however, that $p(a_k^i | Z_{k-1}, A_{k-1}^i)$ could easily be adapted to include prior information based on sensor characteristics or sensor identity.

The factorization in (10) yields a simple algorithm for sampling new particles and, thus, new data assignments, from an existing set of particles. When a new measurement z_k arrives, a new set of particles is sampled as follows.

- 1) For each obstacle in each particle, calculate the likelihood $p(z_k | a_{m,k}^i, Z_{k-1}, A_{k-1}^i) \cdot p(a_{m,k}^i | Z_{k-1}, A_{k-1}^i)$ that the measurement z_k was generated by the m th tracked obstacle in the i th particle, where the notation $a_{m,k}^i$ denotes the event that the i th particle assigns the measurement to its m th of M^i tracked obstacles.
- 2) For each particle, randomly assign z_k to one of its tracked obstacles with probability proportional to the likelihood that the chosen obstacle generated the measurement z_k . In the i th particle, for example, the assignment $a_{m,k}^i$ is chosen with probability proportional to $p(z_k | a_{m,k}^i, Z_{k-1}, A_{k-1}^i) \cdot p(a_{m,k}^i | Z_{k-1}, A_{k-1}^i)$, as calculated in step 1.

Intuitively, sampling is accomplished by asking each particle to expand its history of data assignments from time index $k-1$ to k by randomly assigning the most recent measurement z_k to one of its tracked obstacles. Note that while random, the data assignment decisions are intelligent: As per (10), a tracked obstacle that is more likely to have generated the measurement z_k is more likely to be assigned z_k during sampling.

With a new set of particles drawn according to $q_{\text{opt}}(a_k^i | Z_k, A_{k-1}^i) = p(a_k^i | Z_k, A_{k-1}^i)$, new weights $w_k^i, i \in [1, N]$ must be calculated to preserve the probability density approximation established in (3) and (4). To calculate these new weights recursively from the existing particle weights w_{k-1}^i , the optimal proposal density $q_{\text{opt}}(a_k^i | Z_k, A_{k-1}^i)$ from (10) is substituted as the proposal density $q(a_k^i | Z_k, A_{k-1}^i)$ into (8) to yield

$$w_k^i \propto \frac{1}{\alpha_k^i} w_{k-1}^i \quad (11)$$

where the constant α_k^i is the particle-dependent normalizing constant from (10), and the symbol “ $\propto$ ” indicates that after the weight update, the weights must be renormalized to maintain

- (5). If there is no prior information about the measurement assignment, i.e., $p(a_k^i | Z_{k-1}, A_{k-1}^i) = 1/M^i$ is uniform over the M^i tracked obstacles in the i th particle, then the constant α_k^i for the i th particle is as follows:

$$\alpha_k^i = \left[\frac{1}{M^i} \sum_{m=1}^{M^i} p(z_k | a_{m,k}^i, Z_{k-1}, A_{k-1}^i) \right]^{-1} \quad (12)$$

where $a_{m,k}^i$ is the event that the measurement z_k taken at time index k is assigned to the m th of M^i obstacles in the i th particle. Equation (12) can thus be interpreted as an “average assignment likelihood” over all the tracked obstacles in the i th particle. This weight update has the satisfying interpretation that particles with higher weights contain obstacles that on average are better matches to the sensor measurements. If instead, there were prior assignment information, α_k^i would be a “weighted average assignment likelihood.”

A high-level description of the joint data assignment-tracking algorithm is now given as the following.

- 1) Draw an initial set of particles $A_0^i \forall i \in [1, N]$.
- 2) Predict all obstacles in each particle forward in time to the next measurement z_k to yield a parametric representation of $p(O_k | Z_{k-1}, A_{k-1}^i)$.
- 3) Randomly sample a new set of data assignments for z_k from the optimal proposal density according to (10) and the given sampling procedure.
- 4) For each particle, update the particle’s parametric-tracking filter to yield $p(O_k | Z_k, A_k^i)$, which reflects the random data assignment chosen by the particle.
- 5) Update particle weights according to (11), and renormalize the weights to sum to unity.
- 6) Resample particles to keep the filter well conditioned, if necessary. Effective resampling strategies are discussed in Arulampalam *et al.* [44] and Grisetti *et al.* [46].
- 7) Go to step 2.

Occasionally, the tracking algorithm performs step 6, i.e., a particle resampling step, to alleviate degeneracy problems. These common particle-filtering problems occur over time due to unlucky sampling choices that cause particle weights to degrade asymptotically to zero over repeated renormalizations [44]. In the present implementation, such a problem might arise over time as the particles continue to sample random data assignment decisions without a means of correcting poor choices. If left unchecked, all particles would eventually make poor data assignment decisions, leaving the filter with no particularly good hypothesis among its particles. To avoid this problem, the RBPF’s particles are occasionally resampled into a new set of particles with probability proportional to their weight. Particles with higher weights, which correspond to more likely interpretations of the urban environment, are more likely to be sampled into the new set of particles one or more times. Particles with lower weights, which correspond to unlikely hypotheses, may not even be sampled into the new set of particles at all. To decide when to resample, the algorithm adopts the method proposed by Grisetti *et al.* to estimate the effective number of particles $\tilde{N}_k$ that substantially influence the shape of the filtered density

at time index k [46]

$$\hat{N}_k = \frac{1}{\sum_i (w_k^i)^2}. \quad (13)$$

For normalized particle weights drawn over N particles, (13) implies that $1 \leq \hat{N}_k \leq N$. Intuitively, $\hat{N}_k$ achieves its maximum when all particle weights are equal, which occurs when the particle filter assigns equal likelihood to each of its particles. Similarly, $\hat{N}_k$ achieves its minimum when all but one weight are zero, the undesirable situation in which the particle filter effectively places all likelihood on only one of its hypotheses. Grisetti *et al.* suggest thresholding $\hat{N}_k$ to determine whether resampling is necessary [46]. The present implementation adopts this approach, resampling the particles when $\hat{N}_k \leq N/2$ to make sure that all particles possess comparable weights in the filtered density.

Thus far, it has been assumed that each particle is given a list of existing obstacles O_0^i at $k = 0$ that are all present and visible from the moment the algorithm is first started. In many practical situations, this *a priori* information is not available, and instead, the algorithm must begin with a set of N empty particles: $O_0^i = \emptyset \forall i \in [1, N]$. The creation of new obstacles is then handled within the particle filtered data assignment framework. In particular, a birth likelihood $p(z_k | a_{m=0,k}^i)$ can be defined for each measurement z_k to represent the likelihood that the measurement originates from a newly visible obstacle. The tracker's RBPF then considers this likelihood with those of any existing obstacles when choosing data assignments: The measurement is either assigned to an existing obstacle or used to initialize a new one [47]. In contrast, the algorithm does not remove obstacles from particles with a corresponding death likelihood, as each obstacle's existence depends on its state history as well as its assigned measurements. Obstacle death is, therefore, modeled in the RBPF's parametric estimates of the obstacle density $p(O_k | Z_k, A_k)$, where, for practical reasons, it is implemented as a deterministic decision based on the obstacle's state and measurement history. The convenience of this formulation is that the algorithm automatically determines the number of obstacles in the environment, and obstacles may pass in and out of sensor range without a growing computational burden [47].

B. Continuous Tracking Problem

Section III-A describes the method used in the proposed tracking algorithm to run an RBPF that samples a set of high likelihood data assignment hypotheses. This solution to the data assignment problem relies on the effectiveness of an inexpensive, yet accurate parametric filter for tracking obstacles in the urban environment under known measurement assignments. The goal of this parametric filter is to describe obstacle geometry and motion as accurately as possible to make each obstacle distinct and distinguishable for measurement assignment. This reduces computational effort in the RBPF by decreasing the number of high likelihood data assignment hypotheses and, accordingly, the expected number of particles required to fully sample these hypotheses.

To further reduce the computational burden of the problem, all obstacles are assumed to be conditionally independent, given sensor measurements and their assignments. This allows the continuous tracking problem to be factorized into a set of single-obstacle-tracking problems

$$p(O_k | Z_k, A_k^i) = \prod_m p(O_{m,k} | Z_k, A_k^i) \quad (14)$$

where $p(O_{m,k} | Z_k, A_k^i)$ is the probability density of the m th obstacle in the i th particle. By assuming conditional independence among the obstacles, each particle in the RBPF uses a bank of small parametric filters, one for each obstacle tracked. This factorization offers substantial computational savings, since parametric filters often rely on inexpensive matrix operations with polynomial time complexity. The alternative, i.e., a single large parametric filter that simultaneously estimates the joint state of all obstacles in a particle, would be prohibitively expensive. The parametric filter described in this section, therefore, sets out to track a single obstacle, which is potentially moving, under known measurement assignments.

Since the present implementation is designed to solve the dynamic urban perception problem, all obstacles are assumed to have size and maneuver characteristics similar to full-size motor vehicles. This assumption is made to constrain the algorithm's attention to vehicular traffic, although it does not preclude an expanded list of obstacle classes including pedestrians, bicyclists, tractor-trailers, and other static and dynamic objects that are commonly present in urban environments. The benefit of adopting this assumption is that it avoids an ancillary target classification problem as each obstacle, whether moving or static, is tracked with an instance of the same parametric filter.

The most common parametric filter used for tracking a single full-size moving vehicle models the vehicle as a rectangle [15], [16], [21], [47]. Such a filter typically estimates the vehicle's length and width as well as motion parameters, such as position and velocity. Problems with this approach stem from two sources: The fact that vehicles are not rectangles, and the fact that no sensor can measure the entire shape of the vehicle at once. Unfortunately, these two problems have competing solutions: The rectangle model is not rich enough to describe a moving vehicle, but it is too complex to permit simple Bayesian updates. Many existing approaches strike a compromise by accepting the former problem and applying an *ad hoc* feature extraction or matching algorithm to raw sensor data to address the latter [14]–[16]. However, as MacLachlan and Mertz point out, these feature extraction techniques yield erroneous motion estimates due to the substantial change in the sensed shape of a vehicle as it moves around the sensor's field of view [17].

The proposed tracking algorithm avoids these problems with a filter that operates directly on raw sensor data. The filter contains five state variables describing the position, motion, and shape of the moving vehicle in a coordinate frame fixed to the ego robot. Fig. 1 shows how these state variables define the moving vehicle within the parametric filter. Internally, the filter's tracked vehicle is stored as two pieces of data: a parameterized probability density $p(O_{m,k} | Z_k, A_k^i)$ over the vehicle's state $O_{m,k}$ and a cloud of previously sensed points describing the vehicle's

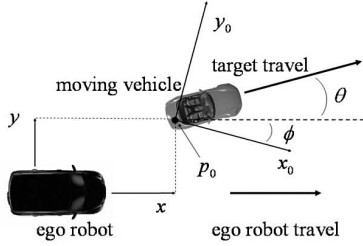


Fig. 1. Coordinate frames used for tracking a single obstacle (a moving vehicle) under known measurement assignments. In the present implementation, an obstacle's position is defined by the x and y location of a reference point p_0 fixed to the obstacle.

shape. Because the point cloud is sensed directly rather than parameterized or estimated, the point cloud representation stores a far more accurate description of the tracked vehicle's geometry than more common rectangular models. A more subtle observation is the fact that integrating the point cloud representation is no more expensive than integrating parameterized and highly simplified vehicle geometries, because the tracked vehicle is a rigid body. As a result, the position and motion of all points in the cloud are related through an unchanging set of affine transformations. If the relative positions of these points are stored in an obstacle-fixed coordinate frame, the motion of one such reference point in the ego robot frame is sufficient to reconstruct the motion of all points. A further key observation is that direct measurements of this one target-fixed reference point, which is labeled p_0 in Fig. 1, are not necessary to accurately estimate the motion of the target vehicle. Instead, the position of the fixed reference point is included in the set of states to be estimated. The resulting tracking filter produces consistent motion estimates by utilizing sensor information within the Bayesian framework to simultaneously estimate the location and motion of an arbitrary fixed reference point on the vehicle being tracked. Because the filter does not rely on measuring a *specific* fixed reference point, such as the tracked vehicle's center of mass, it avoids estimation artifacts that arise from *ad hoc* attempts to locate these points in raw sensor data.

The proposed filter parameterizes a tracked vehicle's position, motion, and shape with five states

$$o_{m,k} = [x_k \ y_k \ s_k \ \theta_k \ \phi_k]^T \quad (15)$$

where, from now on, $o_{m,k}$ will be used instead of $O_{m,k}$ to reflect the desire to estimate the state of the tracked vehicle only at the current time index k . From Fig. 1, the position variables x and y describe the location of the tracked vehicle's arbitrary fixed point relative to the ego robot. The motion of the tracked vehicle's fixed point is parameterized by the velocity variables s and θ , which store the point's absolute ground speed and heading relative to the ego robot. The orientation of the tracked vehicle's obstacle-fixed coordinate frame is parameterized by ϕ , which describes the rigid body rotation angle between the ego robot's coordinate frame and the obstacle-fixed frame in which the tracked vehicle's point cloud is stored. This orientation angle ϕ may differ from the tracked vehicle's relative heading θ , as the axes of the estimated arbitrary obstacle-fixed coordinate

frame do not necessarily align with the direction of the tracked vehicle's motion. Note that if the tracked vehicle's point cloud is stored relative to the fixed point, the combined rotation ϕ and translation $[x \ y]^T$ suffice to locate any point in the point cloud with respect to the ego robot. Similarly, the transformation may be combined with the motion parameters s and θ , and potentially higher order terms, such as acceleration and curvature, to compute the motion of any point in the tracked vehicle's point cloud.

The time evolution of $o_{m,k}$ in the (potentially moving) coordinate frame fixed to the ego robot is modeled by the following continuous-time nonlinear differential equations:

$$\dot{x} = s \cos(\theta) - v_x + y\omega_z + e_x \quad (16)$$

$$\dot{y} = s \sin(\theta) - v_y - x\omega_z + e_y \quad (17)$$

$$\dot{s} = e_s \quad (18)$$

$$\dot{\theta} = -\omega_z + e_\theta \quad (19)$$

$$\dot{\phi} = -\omega_z + e_\phi \quad (20)$$

where v_x and v_y are components of the ego robot's velocity, ω_z is the ego robot's rate of rotation, and e_x , e_y , e_s , e_θ , and e_ϕ are zero mean, mutually uncorrelated, Gaussian, and white random variables acting as process noise. Intuitively, these random variables account for unmodeled maneuvers executed by the tracked vehicle. Note that although (18) assumes the tracked vehicle moves on average in a straight line at constant speed, any parameterized dynamics model may be used to incorporate more advanced maneuvers and motion constraints. This flexibility is a major benefit of the proposed algorithm's point cloud representation: The dynamics model merely describes the time evolution of the rigid body transformation between the coordinate frame fixed to the ego robot and the one fixed to the tracked vehicle.

In addition to the process noise for vehicle maneuvers, it is noted that v_x , v_y , and ω_z are measured by noisy sensors. As a result, the following substitutions are made:

$$v_x = \hat{v}_x + e_{v_x} \quad (21)$$

$$v_y = \hat{v}_y + e_{v_y} \quad (22)$$

$$\omega_z = \hat{\omega}_z + e_{\omega_z} \quad (23)$$

where $\hat{v}_x$, $\hat{v}_y$, and $\hat{\omega}_z$ are measured from odometry sensors on the ego robot, and e_{v_x} , e_{v_y} , and e_{ω_z} are zero mean, mutually uncorrelated, Gaussian, and white random variables acting as additional process noise. These noisy estimates of ego robot motion unfortunately result in correlations between all obstacles tracked, so that (14) is an approximate factorization. In SLAM literature, where obstacles are static and often modeled with no process noise, such correlations are large and central to the localization problem [12]. In a dynamic environment, the opposite is true: Obstacles are modeled with significant uncorrelated process noise to capture uncertainty in maneuvers, such as acceleration and turning. This process noise effectively swamps correlations between obstacles, as uncertainty in their maneuvers is far larger than uncertainty in odometry

sensors. As a result, the factorization in (14) is taken as a valid approximation.

To facilitate the use of a computationally inexpensive parametric filter, $p(o_{m,k}|Z_k, A_k^i)$ is assumed to be Gaussian. In light of the weak nonlinearities in (16)–(20) for full-size vehicles driving in traffic, the time evolution of this Gaussian is computed in the tracking algorithm using the prediction step of the extended KF (EKF). The prediction step is implemented using a fourth-order Runge–Kutta numerical integration to convert (16)–(20) to a nonlinear discrete time difference equation of the form

$$o_{m,k+1} = f(o_{m,k}, v_{m,k}) \quad (24)$$

where $v_{m,k}$ is a vector of zero mean, mutually uncorrelated, Gaussian, and white random variables derived from e_x , e_y , e_s , e_θ , e_ϕ , e_{v_x} , e_{v_y} , and e_{ω_z} . Traditional EKF equations are then used to compute the time evolution $p(o_{m,k+1}|Z_k, A_k^i, \hat{v}_{x,k}, \hat{v}_{y,k}, \hat{\omega}_{z,k})$ conditioned on past measurements, assignments, and the measured motion of the ego robot [1]. Although conditioning on measurements of ego robot motion is henceforth suppressed for brevity, it is understood to be present in the time evolution of all obstacles.

When a new sensor measurement z_{k+1} is assigned to the tracked vehicle, the vehicle's posterior probability density can be updated to $p(o_{m,k+1}|Z_{k+1}, A_{k+1}^i)$ to reflect the new measurement. The exact form of the update depends greatly on the information contained in the measurement and, therefore, on the type of sensor generating the measurement. Three popular automotive sensing modalities are explored for the present implementation: laser rangefinder, radar, and optical camera. Measurement updates for these sensing modalities are discussed in Section III-B1 and 2.

1) *Laser Rangefinder*: The first type of sensor fused in the proposed tracking algorithm is the laser rangefinder, which measures a point cloud of returns generated from patches in the environment reflecting the sensor's energy emitted in one or more scanning beams. Although laser rangefinders offer centimeter-level ranging and subdegree bearing accuracy, a practical problem arises due to the fact that they measure individual points rather than entire objects. The existing approaches solve the problem by extracting features from the point cloud, such as lines, corners, rectangles, or the center of mass and then using these features as measurements to update the parameterized probability density of the tracked vehicle [14]–[18]. Unfortunately, these feature extraction algorithms are unstable in even mildly dynamic environments, where the shape and motion of tracked vehicles and pitch and roll of the ego robot cause rapid changes in object shape. Unmodeled instability in these features frequently yields erroneous state estimates, particularly in the motion of the tracked vehicle. Fig. 2 illustrates the problem by plotting the locus of point cloud centers of mass observed as a moving vehicle passes in front of a stationary laser rangefinder. Here, the vehicle's motion causes the observed center of mass to drift with respect to the true center of mass. On average, the observed center of mass lies more than 2 m from the true center of mass, and all measurements are at least 60 cm from the truth. Numeric indices of center of mass measurements in Fig. 2 also

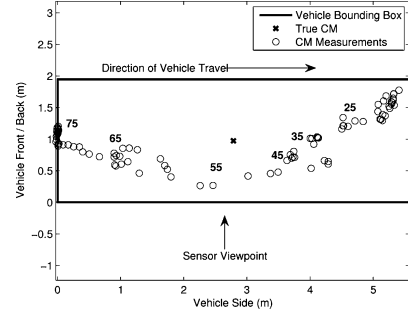


Fig. 2. Locus of point cloud centers of mass observed as a moving vehicle passes in front of a stationary laser rangefinder. The apparent drift in the observed center of mass results from dramatic changes in the sensed shape of the moving vehicle as it passes in front of the laser rangefinder. This sensitivity, appearing in even mildly dynamic environments, causes instability in feature extraction algorithms, such as center of mass, edge, and corner detectors applied to laser rangefinder data.

show that the motion of the observed measurements is largely opposite from the vehicle's direction of travel.

Two alternatives exist to alleviate estimation problems stemming from instability in sensed features. The first alternative is to model the instability, by accounting for the extra uncertainty in the measurement likelihood $p(z_{k+1}|o_{m,k+1}, A_{k+1}^i)$. While this solution may yield consistent state estimates, it is unsatisfactory in the sense that the resulting estimation errors are larger to account for measurement instability. The second alternative, which has been adopted for laser rangefinder data fused in the present implementation, is to use an alternate set of measurements that are more reliably stable. First, the laser points are grouped into distinct objects via a clustering algorithm. The conservative algorithm presented in Miller *et al.* [29] yields stable performance in practice and is used here, although the conservative algorithm presented in Miller and Campbell [47] could also be used. After the laser points are clustered into distinct objects, three measurements are extracted: the smallest and largest bearings $b_{\min}$ and $b_{\max}$ from each cluster, and the range $r_{\min}$ to the closest point in each cluster. A measurement vector z_{k+1} consists of these three measurements extracted from a single cluster

$$z_{k+1} = [b_{\min} \quad b_{\max} \quad r_{\min}]^T \quad (25)$$

and depending on the number of clusters present, multiple measurement vectors may be extracted from a single frame of laser returns.

The most important aspect of the bearing–bearing–range measurement described in (25) is its stability: The values of $b_{\min}$, $b_{\max}$, and $r_{\min}$ change slowly as the tracked vehicle's point cloud undergoes small transformations. This property is not shared by other extracted features, such as the point cloud's center of mass, which may suffer large discontinuities in the face of small transformations. Such strong nonlinearities make these features unsuitable for linear estimation techniques.

In contrast, the weak nonlinearities in the bearing–bearing–range measurement make it ideal for use in estimation techniques, which linearize the relationship between the measurement and state vector. For the present implementation, this is

made explicit with the following auxiliary variables:

$$\begin{aligned}\beta_{\min} &= \min_{c \in C_{m,k+1}} \angle [T_s(x, y, \phi)(c - l_s)] \\ \beta_{\max} &= \max_{c \in C_{m,k+1}} \angle [T_s(x, y, \phi)(c - l_s)] \\ \rho_{\min} &= \min_{c \in C_{m,k+1}} \|T_s(x, y, \phi)(c - l_s)\| \end{aligned} \quad (26)$$

where $\beta_{\min}$, $\beta_{\max}$, and $\rho_{\min}$ constitute the state-generated measurement, $C_{m,k+1}$ is the set of points corresponding to the m th tracked vehicle's point cloud stored from previous sensor measurements, $T_s(\cdot)$ is the transformation matrix that projects the point cloud $C_{m,k+1}$ into the coordinate frame of the laser rangefinder, l_s is the location of the laser rangefinder, and the operators $\angle(\cdot)$ and $\|\cdot\|$ return the bearing and magnitude of their vector arguments, respectively. With these auxiliary variables defined, the bearing–bearing–range measurement in (25) relates to the m th tracked vehicle's state through the nonlinear measurement function $h_L(\cdot)$

$$h_L(o_{m,k+1}, C_{m,k+1}) = \begin{bmatrix} \beta_{\min} \\ \beta_{\max} \\ \rho_{\min} \end{bmatrix}. \quad (27)$$

With the measurement function defined, the measurement likelihood $p(z_{k+1}|o_{m,k+1}, A_{k+1}^i)$ is modeled as Gaussian, corrupted by additive zero-mean Gaussian white noise w_{k+1}

$$z_{k+1} = h_L(o_{m,k+1}, C_{m,k+1}) + w_{k+1} \quad (28)$$

where the Gaussian measurement noise w_{k+1} is modeled as additive to reflect the fact that the laser rangefinder measures bearings and ranges directly. With $p(z_{k+1}|o_{m,k+1}, A_{k+1}^i)$ modeled as Gaussian, linear measurement updating techniques can be used to update $p(o_{m,k+1}|Z_k, A_{k+1}^i)$ to $p(o_{m,k+1}|Z_{k+1}, A_{k+1}^i)$ to reflect the new measurement. In particular, the algorithm utilizes the sigma point transform in the update step of the SPF, because numerical differentiation of $h_L(o_{m,k+1}, C_{m,k+1})$ is more convenient than explicit differentiation. This measurement is an ideal complement to the point cloud representation of the tracked vehicle, as neither rely on knowledge of a particular fixed point on the tracked vehicle. Instead, information about the tracked vehicle's motion and evolving rigid body transformation are gathered indirectly from robust measurements via fusion in a Bayesian tracking filter.

2) *Radar and Optical Camera*: The second and third types of sensors fused in the proposed tracking algorithm are radar and optical cameras. Like the laser rangefinder, measurements from these sensors are fused in the algorithm at the object level. In other words, the algorithm relies on external processing to group raw sensor data into measurements of distinct objects. In the case of radar and optical camera data, this processing is commercially available; radar and camera systems built for collision detection often process raw data into objects. For the radar, the algorithm utilizes measurements of the tracked vehicle's bearing, range, and range rate in the sensor's coordinate frame

$$z_{k+1} = [b_s \quad r_s \quad \dot{r}_s]^T \quad (29)$$

and as with (25), multiple measurements may be available if more than one vehicle is present.

Much like the laser rangefinder, the radar measurement suffers from ambiguity. The radar's range and bearing components do not measure a particular fixed point on the tracked vehicle, and in fact, the measured point may change over time as reflective portions of the vehicle become visible or occluded. The difficulty is further compounded by the fact that the radar's algorithm for generating bearing and range measurements is often unknown, as many off-the-shelf radar units do not document their internal measurement processing algorithms. As a result, there is no way to relate the radar's measurements to the tracked vehicle's point cloud to eliminate the measurement ambiguity completely. Instead, the tracked vehicle's point cloud is used to generate an approximate measurement function $h_R(o_{m,k+1}, C_{m,k+1})$ for the radar

$$h_R(\cdot) = \begin{bmatrix} \frac{1}{2}(\beta_{\min} + \beta_{\max}) \\ \rho_{\min} \\ \{v_v(s, \theta) - v_s(\hat{v}_x, \hat{v}_y, \hat{\omega}_z)\} \cdot e_r(x, y) \end{bmatrix} \quad (30)$$

where $v_v(\cdot)$ is the velocity of the tracked vehicle, $v_s(\cdot)$ is the velocity of the sensor on the ego vehicle, and $e_r(\cdot)$ is the unit vector from the radar to the tracked vehicle. As with the laser rangefinder, the measurement likelihood $p(z_{k+1}|o_{m,k+1}, A_{k+1}^i)$ is modeled as Gaussian, corrupted by additive zero-mean Gaussian white noise w_{k+1}

$$z_{k+1} \approx h_R(o_{m,k+1}, C_{m,k+1}) + w_{k+1} \quad (31)$$

where, in approximation, the radar measures the center bearing, the closest range, and the range rate of the tracked vehicle's arbitrary fixed point. The ambiguity of the radar measurement is then addressed in the covariance matrix of the measurement noise w_{k+1} , where bearing and range measurement noise standard deviations are set large enough to account for the fact that the measurements can correspond to any point on the tracked vehicle, and for the fact that the true measured point may have a slightly perturbed velocity if the tracked vehicle is turning. As with the laser rangefinder, the sigma point transform of the SPF update is then used to generate $p(o_{m,k+1}|Z_{k+1}, A_{k+1}^i)$ from the radar measurement.

Optical camera measurements are incorporated into $p(o_{m,k+1}|Z_k, A_{k+1}^i)$ similar to radar measurements. For the optical camera, the algorithm utilizes measurements of the tracked vehicle's centerline (axis of symmetry), range rate, and width in the camera's image coordinate frame

$$z_{k+1} = [r_s \cos(b_s) \quad r_s \sin(b_s) \quad \dot{r}_s \quad w_s]^T \quad (32)$$

where r_s is the range to the closest point on the tracked vehicle's centerline as observed in the camera's image coordinate frame, b_s is the bearing of that centerline point, $\dot{r}_s$ is the tracked vehicle's range rate, and w_s is the width of the tracked vehicle as detected in the image frame.

Unlike the laser rangefinder and radar, the optical camera measurement is not necessarily ambiguous. If edge or symmetry kernels are used to find the bearing of the tracked vehicle's centerline as observed from the front or rear, then the measurement is not ambiguous. In fact, the sensed bearing b_s is the average of the minimum and maximum bearing measurements



Fig. 3. Proposed tracking algorithm is implemented in real time on Cornell University's "Skynet," which is an autonomous 2007 Chevrolet Tahoe that completed the 2007 DUC.

introduced in (25):

$$b_s = b_{\text{avg}} = \frac{1}{2} (b_{\min} + b_{\max}). \quad (33)$$

In other words, the tracked vehicle's centerline lies halfway between its two extreme edges. Similarly, the model-driven centerline bearing β_{avg} from the point cloud representation is as follows:

$$\beta_{\text{avg}} = \frac{1}{2} (\beta_{\min} + \beta_{\max}). \quad (34)$$

Using β_{avg} and the other stable measurements, the optical camera's measurement function $h_O(o_{m,k+1}, C_{m,k+1})$ is as follows:

$$h_O(\cdot) = \begin{bmatrix} \rho_{\min} \cos(\beta_{\text{avg}}) \\ \rho_{\min} \sin(\beta_{\text{avg}}) \\ \{v_v(s, \theta) - v_s(\hat{v}_x, \hat{v}_y, \hat{\omega}_z)\} e_r(x, y) \\ 2\rho_{\min} \tan\left(\frac{1}{2} \{\beta_{\max} - \beta_{\min}\}\right) \end{bmatrix}. \quad (35)$$

Note that $\rho_{\min}$ is used as an approximate measure of the range to the tracked vehicle's centerline. This approximation is valid in the sense that the optical camera cannot accurately resolve range (depth) in the image frame. Using the measurement function, the camera measurement likelihood $p(z_{k+1} | o_{m,k+1}, A_{k+1}^i)$ is also modeled as Gaussian, corrupted by additive zero-mean Gaussian white noise w_{k+1}

$$z_{k+1} = h_O(o_{m,k+1}, C_{m,k+1}) + w_{k+1}. \quad (36)$$

The sigma point transform of the SPF update is used to generate $p(o_{m,k+1} | Z_{k+1}, A_{k+1}^i)$ from the camera measurement.

IV. EXPERIMENTAL PERFORMANCE

The proposed tracking algorithm has been implemented in real time on Cornell University's "Skynet," which is an autonomous 2007 Chevrolet Tahoe shown in Fig. 3. Skynet is equipped with seven laser rangefinders: three quadruple-laser Ibeo ALASCA XTs in the front bumper, one single-laser SICK LMS 220 in the back bumper, two single-laser SICK LMS 291s in the rear driver and passenger doors, and a 64-laser Velodyne HDL-64E on the roof. The placement and coverage of these laser rangefinders is shown in Fig. 4. Skynet is also equipped with eight Delphi forward-looking radar system (FLR) millimeter-wave radar units: five in the front bumper and three in the back bumper. The placement and coverage of these radars is shown in

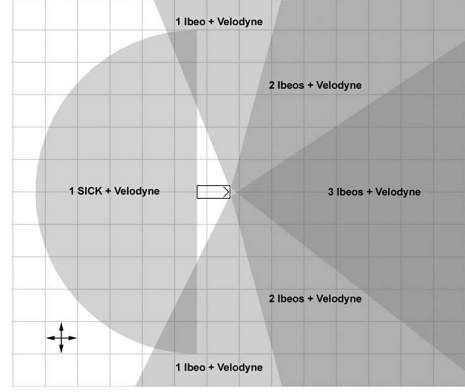


Fig. 4. Sensor placement and coverage diagram for Skynet's laser rangefinders. Skynet sits in the center of the diagram, facing right. Redundant laser rangefinders cover the front of Skynet's field of view, where it encounters the most dangerous moving obstacles.

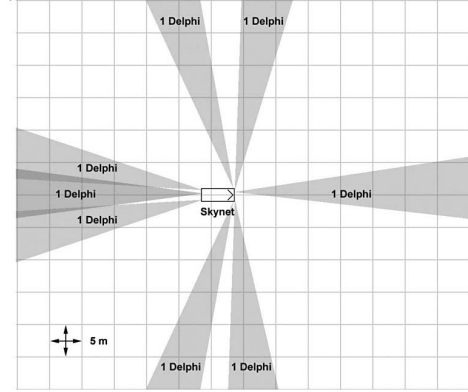


Fig. 5. Sensor placement and coverage diagram for Skynet's radars. Skynet sits in the center of the diagram, facing right. Radars are placed to detect oncoming cars in opposing lanes, from the left and right in merging situations, and from the rear in passing situations.

Fig. 5. Finally, Skynet is equipped with two optical cameras: a forward-facing Basler A622F and a backward-facing Unibrain Fire-i 520b, both mounted on the roof and running MobilEye SeeQ vehicle-tracking software. All sensors are accurately time-stamped via synchronized microcontroller interfaces, and each is available for obstacle detection and tracking [29]. Skynet is also equipped with a position, velocity, and attitude estimator that fuses GPS with onboard vehicle odometry and inertial navigation to generate accurate and robust localization, as well as differential vehicle motion estimates [48].

The algorithm has been implemented in C++ and connected to Skynet's time-stamped data logs. In this practical implementation, the algorithm is initialized with no prior information. A birth likelihood is then used to discover new obstacles in the RBPF framework as per Section III-A. This sensor-dependent birth likelihood is uniform over the corresponding sensor's field of view, except that it tapers to zero near existing tracked obstacles. This allows new obstacles to appear anywhere in the sensor's field of view, except near existing tracked obstacles, which are all assumed to be solid. Similar to the birth model, each

measurement is also evaluated against a clutter model created for each of Skynet's sensors. The corresponding clutter likelihood captures common sensor errors with uniform, Gaussian, and multimodal Gaussian densities that account for multiple reflections, signal multipath, and false positives in each sensor. Deletion of old obstacles is similarly tied to the Bayesian framework but implemented with deterministic thresholds to guard against the danger of randomly deleting a threatening obstacle. Each obstacle's existence is modeled as a probability that decays exponentially to zero with a 95% time constant of 3 s. Each measurement assigned to an obstacle provides evidence, in a Bayesian sense, for that obstacle's existence, and an occupancy grid created from Skynet's Velodyne provides evidence against false positives. Obstacles are removed from the RBPF's particles when their existence probability drops below 5%.

Specific practical accommodations are also made to adapt the point cloud representation across the algorithm's three sensing modalities. Since neither the Delphi radars nor the MobilEye SeeQ software generate point clouds, obstacles that have not yet been assigned laser rangefinder measurements are tracked as single-point obstacles with parametric filters only, i.e., with no point clouds. Laser rangefinder measurements are segmented via the clustering algorithm discussed in Miller *et al.* [29]. This algorithm generates clusters only when statistically distinct minimum and maximum bearings can be observed in the data; any remaining unclustered data is treated as part of the static background. A further hypothesis test is applied to reject clusters, whose width (given sensor noise characteristics) is so large or so small that they could not possibly represent a moving object. This test was originally based on DUC rules that placed limits on competitor size. It is retained here as *a priori* knowledge about the sizes of objects that can move in the real world. These two tests improve the data's resilience to the two common clustering errors: breaking a single object into multiple small clusters and grouping unrelated objects into a single large cluster. In addition, new point clouds assigned to an obstacle always overwrite existing point clouds, but only after the measurement update is performed in the obstacle's parametric filter. In other words, no attempt is made to merge multiple point clouds over time; only the obstacle's rigid body transform and motion parameters undergo Bayesian estimation. Two reasons support this design choice: The accuracy and scanning rates of laser rangefinders make it unnecessary to combine multiple point clouds, and the computational burden required to store and process increasingly large point clouds would quickly overwhelm any real-time implementation.

The proposed algorithm is tested in a series of three experiments. The first two experiments evaluate the tracking capabilities of the system in common but difficult maneuvers: a perpendicular intersection encounter similar to the circumstances of Fig. 2 and a parallel head-on encounter. In these experiments, the system is evaluated against truth data obtained from pose estimators on board the ego vehicle and the target vehicle being tracked. These estimators fuse traditional GPS and differential corrections with inertial navigation sensors on board both vehicles to produce position, velocity, and attitude estimates for both the ego and target vehicle with submeter position accuracy, centimeter-

per-second-level velocity accuracy, and subdegree attitude accuracy [48]. The system simultaneously tracks other obstacles in the environment aside from the target vehicle, although ground truth for these obstacles is unavailable and, therefore, not evaluated. These experiments are performed at the Seneca Army Depot in Romulus, NY, which contains a variety of features, including paved and unpaved roads, some painted road lines, potholes, railroad tracks, short and tall vegetation, and buildings.

In the first two experiments, the primary goal is to show that the system can accurately track a moving target vehicle from a potentially moving ego vehicle without estimation artifacts. The criteria used to evaluate the system are, therefore, average estimation errors in metrics critical to dynamic obstacle avoidance: range to the target vehicle's closest point, the target vehicle's ground speed, and the target vehicle's relative heading. In addition to average estimation errors, statistical hypothesis tests are conducted on the distributions of observed estimation errors to evaluate biases in these errors that might be indicative of estimation artifacts. These hypothesis tests are conducted as paired *t*-tests, which evaluate whether the sampled solution estimated by the proposed tracking algorithm matches the sampled true solution recovered from the target vehicle and ego vehicle pose estimators. For comparison, the algorithm's error statistics are evaluated against the performance of an EKF estimating the location of the target vehicle's center as if the target were a point. This EKF uses the same process noise parameters as the proposed tracking algorithm, but it performs updates using the center of mass of laser rangefinder clusters, the range, bearing, and range rate of radar returns, and the center of mass of camera tracks as if they corresponded to the center of the target vehicle. In other words, the EKF uses the *ad hoc* features discussed in Section III-B1 and Fig. 2. The EKF operates under correct data assignments generated from the truth data and is given perfect knowledge of the target vehicle's true length and width. The goal will be to show that the *ad hoc* EKF yields significant estimation artifacts even with this unrealistic truth information, whereas the proposed algorithm produces unbiased target estimates without it. Additionally, the first two experiments have been designed so that linear combinations of these experiments cover all possible target vehicle approach vectors.

The third experiment evaluates the system's consistency in tracking multiple obstacles in a densely populated, highly dynamic environment. In this experiment, variants of the proposed algorithm with different numbers of particles are run on the same segment of logged data, a 19-min excerpt of a DUC qualifying round at George Air Force Base in Victorville, CA. The data contains Skynet's sensor measurements of two concentric loops of heavy traffic traveling in opposite directions, recorded as Skynet merged into and out of this traffic multiple times. No truth data is available for the qualifying round; therefore, consistency among the algorithm variants is used to evaluate their performance.

A. Experiment 1: Perpendicular Intersection Encounter

The first experiment evaluates the algorithm's ability to track a moving target vehicle as it crosses the ego vehicle's path at

Ex.-1019 - ADB Safegate et al.

ADB Safegate et al. v. Honeywell International Inc. – PGR2023-00053

Page 12



Fig. 6. In the first experiment, the proposed tracking algorithm tracks a moving target vehicle as it crosses the ego vehicle's path at an intersection.

an intersection. In this experiment, the ego vehicle (Skynet) remains stationary at the intersection, while the target vehicle drives past at approximately 15 m/h. The algorithm is run on the collected sensor data with 50 particles, and each particle is allowed to track as many obstacles as it sees fit. Data is collected for a total of 22 perpendicular intersection encounters: 11 with the target vehicle approaching from the right and 11 with the target vehicle approaching from the left. Fig. 6 shows the experimental setup.

When evaluating the system's tracking performance on this data, comparisons are made to the algorithm's maximum a posteriori (MAP) estimate of the environment, which is the particle with the largest weight.

Within this most likely particle, evaluations are made with the tracked obstacle that most closely matches the truth data. This selection step distinguishes the target vehicle from other tracked obstacles of potential interest. The best-matching obstacle is chosen according to a minimum weighted L_2 -norm criterion combining minimum range and minimum and maximum bearing. Weights are chosen such that a simultaneous range error of 5 m and bearing errors of 5° yield unit distance. Any iterations in which no tracked obstacle has a weighted L_2 -norm ≤ 7.8147 , which corresponds to a 95% confidence bound on a χ^2 random variable with three degrees of freedom, are considered missed detections and discarded. A total of 11 such missed detections occurred over the 28 100 algorithm iterations considered in this experiment.

Proposed algorithm and *ad hoc* tracking statistics averaged over the 11 from-the-right encounters are shown in Figs. 7–9. In each figure, statistics are parameterized by the true bearing of the target vehicle's front bumper to align the 11 trials, despite minor variations in maneuver duration. The figures read left to right: The target vehicle approaches the intersection at negative bearings, crosses the ego vehicle's path at bearing zero, and departs the intersection at positive bearings. The target vehicle is closest to the ego vehicle near bearing zero. Fig. 7 plots average tracking errors for the proposed algorithm and the *ad hoc* tracker in range to closest point on the moving target vehicle observed in the 11 intersection encounters. Predictably, ranging errors in the proposed algorithm are smallest in the middle of each encounter when the target vehicle is within 40 m (between -75° and 77°) of the ego vehicle and visible both by side-facing radars and laser rangefinders. Ranging errors increase between 0° and 40° , where the target vehicle is observed by Skynet's forward-facing radar. This radar occasionally pro-

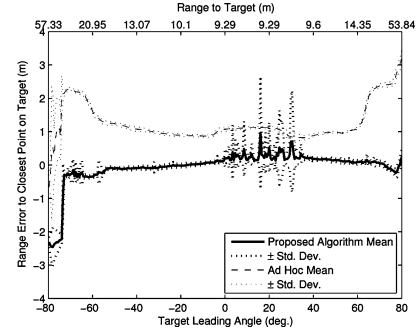


Fig. 7. Range tracking errors to the closest point on a moving target vehicle in the proposed algorithm and an *ad hoc* tracker over 11 perpendicular intersection encounters with the target approaching from the right at 15 m/h. Error statistics are calculated across encounters according to the true bearing of the target's leading edge. Statistically, the proposed algorithm's range estimation errors are largely unbiased, whereas a filter using *ad hoc* measurements yields significant range biases due to changes in the target vehicle's perceived shape and unmodeled measurement errors.

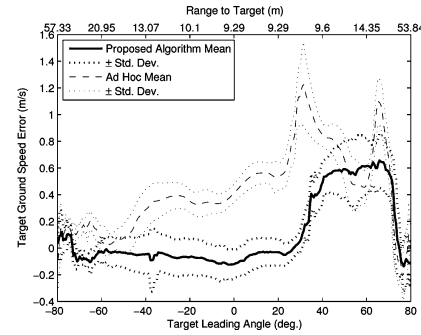


Fig. 8. Ground-speed tracking errors in the proposed algorithm and an *ad hoc* tracker over 11 perpendicular intersection encounters with a moving target vehicle approaching from the right at 15 m/h.

vides erroneous information during perpendicular intersection encounters, as the target vehicle only travels perpendicular to its radial direction. These significantly incorrect measurements yield larger errors temporarily in one or two encounters, resulting in larger sample standard deviations. Errors in the proposed algorithm are largest at the beginning of the maneuver when the target vehicle is first observed between 60 and 80 m from the ego vehicle. In contrast, the *ad hoc* tracking scheme produces range estimates with significant biases due to unmodeled measurement errors based on perceived changes in target shape. These biases are significant throughout the run, despite accurate sensor information present when the target vehicle is in close proximity to the ego vehicle. Paradoxically, these points of closest approach are also the most critical for safe robotic driving. Statistically, the proposed algorithm's range estimation errors have less than a 20 cm bias at the 5% significance level in 251 of the 284 target bearings considered. In contrast, the *ad hoc* tracker's estimation errors achieve the same level of bias in only five of the 284 target bearings.

Fig. 8 plots average target vehicle ground speed estimation errors for the proposed algorithm and the *ad hoc* tracker over the 11 from-the-right perpendicular intersection encounters. The

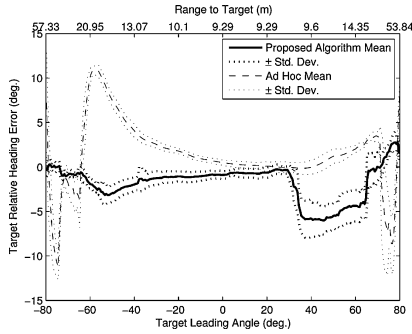


Fig. 9. Relative heading tracking errors in the proposed algorithm and an *ad hoc* tracker over 11 perpendicular intersection encounters with a moving target vehicle approaching from the right at 15 m/h. The proposed algorithm's point cloud representation is able to estimate relative heading correctly despite significant changes in the target vehicle's perceived shape.

proposed algorithm's speed estimation errors remain low from first acquisition through its approach, showing its ability to combine accurate radar speeds with accurate laser ranges to produce an obstacle estimate accurate in both speed and position. Accurate speeds are maintained even as the target vehicle crosses in front of the ego vehicle and out of view of the radars. Small speed errors are incurred temporarily as the target vehicle departs the intersection, where Skynet's left Ibeo laser rangefinder is the primary source of laser data. These errors are a result of an angular miscalibration in the laser rangefinder, resulting in disagreement between Skynet's left and center laser rangefinders. Since the disagreement is in sensor yaw, it creates an unmodeled bias in the evolution of the target vehicle's point cloud, while minimally affecting Skynet's estimates of range. The errors are reduced at bearings near 80° , when it is too far away for the laser rangefinders to produce usable clusters. Despite the benefit of perfect data assignments, the *ad hoc* tracker still maintains significant biases in speed estimates throughout each pass. Statistically, ground speed estimation errors in the proposed algorithm have no significant bias at the 5% significance level in 142 of 284 target bearings considered. At the same significance level, ground speed estimation errors in the proposed algorithm have less than a 0.1 m/s bias in 211 of 284 target bearings. In contrast, the *ad hoc* tracker achieves no significant bias in only 22 bearings and less than a 0.1 m/s bias in only 66 bearings.

Fig. 9 plots average target vehicle relative heading estimation errors for the proposed algorithm and the *ad hoc* tracker over the 11 from-the-right perpendicular intersection encounters. As with the ground speed estimation errors, the proposed algorithm's target vehicle relative heading estimation errors remain relatively small from first acquisition through the intersection. It is noted, however, that these errors appear slightly biased due to a minor angular miscalibration in the ego vehicle's right-facing laser. Estimated relative headings also suffer larger errors at target vehicle bearings between 40° and 80° due to Skynet's miscalibrated laser rangefinder. The *ad hoc* tracker does not suffer as much from these miscalibrations, as truth data has been used to throw out the worst of the measurements during data assignment. This lack of data does, however, result in large



Fig. 10. In the second experiment, the proposed algorithm tracks a moving target vehicle as it approaches the moving ego vehicle from the opposite direction. Both vehicles travel at approximately 15 m/h for these parallel head-on encounters.

heading errors in the *ad hoc* tracker near the beginning and end of each encounter, where the tracker relies heavily on radar measurements. Statistically, biases in the proposed algorithm's heading errors are small: The system's heading estimation errors are biased by less than 2° at the 5% significance level in 210 of the 284 target bearings considered. Despite access to truth data for perfect data assignments, the *ad hoc* tracker is still worse; it achieves biases less than 2° in only 182 target bearings.

B. Experiment 2: Parallel Head-On Encounter

The second experiment evaluates the ability to track a moving target vehicle from a moving ego vehicle as they approach each other from opposite directions on parallel tracks. In this experiment, the ego vehicle (Skynet) and the target vehicle approach each other in opposite lanes on a straight road. Both vehicles travel at approximately 15 m/h, for a combined closing speed of approximately 30 m/h. The proposed algorithm is run with 50 particles, and each particle is allowed to track as many obstacles as it deems appropriate. Data is collected for 11 parallel head-on encounters, all conducted on the same road from the same vehicle starting positions. Fig. 10 shows the experimental setup.

Comparisons of the head-on encounters are made to the obstacles tracked in the RBPF's MAP estimate of the environment. The same L_2 -norm criterion is used to choose the obstacle within this MAP estimate that most closely matches the truth data, and the same L_2 -norm threshold is applied to discard frames in which the system was not tracking the target vehicle. The system experienced 231 of these missed detection frames among the 7264 considered in this experiment. All but one of these missed detection frames occurred when the target vehicle was more than 95 m from the ego vehicle.

Tracking statistics for the proposed algorithm and the *ad hoc* tracker averaged over the 11 parallel head-on encounters are shown in Figs. 11–13. In this experiment, tracking data from the trials is aligned by the true range to the closest point on the target vehicle. Each figure, therefore, reads right to left; the target vehicle enters detection range at approximately 120 m and approaches the ego vehicle in the oncoming lane until both pass each other. Evaluation of the maneuver ends when the range to the closest point on the target vehicle is at a minimum, approximately 3 m. At this point, the target enters a sensing

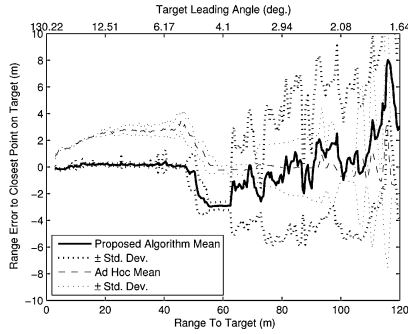


Fig. 11. Range tracking errors to the closest point on a moving target vehicle in the proposed algorithm and an *ad hoc* tracker over 11 parallel head-on encounters at 30 m/h closing speeds. Error statistics are calculated across encounters according to the true range to the target vehicle. Statistically, the proposed algorithm's range estimation errors have less than a 20 cm bias at the 5% significance level in 153 of 284 true target ranges considered. In contrast, a filter using *ad hoc* measurements yields significant range biases due to changes in the target vehicle's perceived shape.

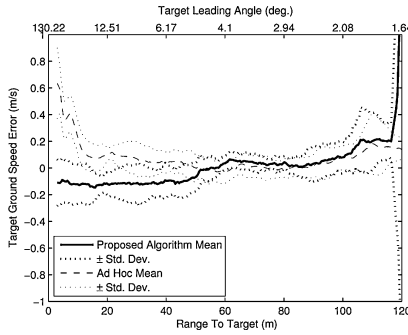


Fig. 12. Ground-speed tracking errors in the proposed algorithm and an *ad hoc* tracker over 11 parallel head-on encounters at 30 m/h closing speeds.

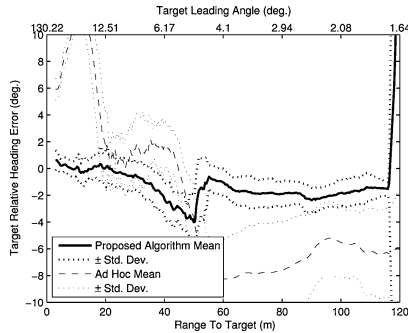


Fig. 13. Relative heading tracking errors in the proposed algorithm and an *ad hoc* tracker over 11 parallel head-on encounters at 30 m/h closing speeds.

blind spot just off Skynet's driver's side, as shown in Figs. 4 and 5. Occasionally, the target persists through this blind spot and is tracked by Skynet's rear sensors, although often the target is deleted and must be reinitialized.

Fig. 11 plots average tracking errors in range to closest point on the moving target vehicle observed in the 11 parallel head-on encounters with the proposed algorithm and the *ad hoc* tracker. The proposed algorithm's range errors are largest when the maneuver begins and when only a single radar ob-

serves the target vehicle. The proposed algorithm's range errors drop to near zero at ranges of approximately 50 m, when the target vehicle is first observed by laser rangefinders. The system also becomes more accurate and precise with the additional sensor data at these ranges, as shown by its smaller errors and standard deviations across the trials. *Ad hoc* tracking errors yield similar performance at long range, where the range error in the radar is large enough to overwhelm the effects of estimation artifacts. At closer ranges, the estimation artifacts become significant; *ad hoc* estimation errors exceed 2 m as the target vehicle approaches the ego vehicle. Paradoxically, the unmodeled measurement errors yield the worst estimation artifacts at mission critical times, where the target vehicle is close to the ego vehicle. Statistically, range estimation errors in the proposed algorithm have less than a 20 cm bias at the 5% significance level in 153 of 284 true target ranges considered. The *ad hoc* tracker performs substantially worse, achieving similar statistical biases in only 134 true target ranges.

Fig. 12 plots average target vehicle ground speed estimation errors in the proposed algorithm and the *ad hoc* tracker over the 11 parallel head-on encounters. Like the perpendicular intersection encounters, the algorithm's ground speed estimates in the head-on encounters remain accurate at all ranges of the maneuver after initial target acquisition. As in the previous experiment, this accuracy is due to the positioning of the ego vehicle's Delphi radars; one faces forward and measures the speeds of approaching vehicles directly through range rate information in Doppler shifts. Of great interest is the fact that these accurate speed estimates persist in the closest ranges of the maneuver, as the target vehicle passes to the left of the ego vehicle. At this point in the maneuver, the target vehicle is not visible by the forward-facing radar, and its shape as observed by the laser rangefinders varies greatly due to a rapidly changing perceived shape. The proposed algorithm's point cloud representation suffers no substantial losses from the changing viewpoint, however, and maintains accurate estimates throughout the maneuver. The *ad hoc* tracking algorithm suffers significant estimation biases, particularly when the target vehicle is closest to the ego vehicle. This catastrophic behavior results from the rapid changes in the target vehicle's perceived shape as it passes by the ego vehicle at the end of the maneuver. Statistically, ground speed estimation errors in the proposed algorithm have no significant bias at the 5% significance level in 105 of 284 true target ranges considered. At the same significance level, ground speed estimation errors in the proposed algorithm have less than a 0.05 m/s bias in 246 of 284 true target ranges. The *ad hoc* tracker only achieves a 0.05 m/s or less statistical bias in 231 true target ranges.

Fig. 13 plots average target vehicle relative heading estimation errors in the proposed algorithm and the *ad hoc* tracker over the 11 parallel head-on encounters. Like the ground speed estimation errors, the proposed algorithm's relative heading estimation errors remain small throughout the entire maneuver. A "notch" is observed near 50 m, however, at ranges where the target becomes reliably visible by the laser rangefinders. The notch is present, while the tracker propagates target shape and position corrections through the estimator's covariance matrix. At the 5% significance level, these heading estimation



Fig. 14. In the third experiment, variants of the proposed algorithm are run on excerpts of data from a DUC qualifying round. In this data, Skynet autonomously completes multiple merges into and out of moving traffic across a lane of oncoming vehicles.

errors have no significant bias in 93 of 284 true target ranges considered. Similarly, the heading estimation errors have less than a 2° bias in 266 of 284 true target ranges. Like the ground speed errors, the proposed algorithm's heading errors remain accurate and largely unbiased, even in the closest ranges of the maneuver, when the target vehicle's shape, as observed by the laser rangefinder, changes most rapidly. The system's point cloud representation and parameterized rigid body transform resolve these rapid shape changes correctly and without estimation artifacts present in feature extraction approaches. In contrast, the *ad hoc* tracker's estimation errors possess significant bias near the end of each run, due to the rapid change in the target vehicle's perceived shape. As with many of the other errors the *ad hoc* tracker incurs, this error is worst at the most critical times: when the target vehicle is closest to the ego vehicle. Statistically, the *ad hoc* tracker's heading estimation errors have less than a 2° bias in only 231 of 284 true target ranges.

C. Experiment 3: Multiple Obstacle Tracking in Heavy Traffic

The third experiment evaluates the proposed algorithm's consistency in tracking multiple obstacles in a densely populated, highly dynamic environment. In this experiment, variants of the proposed algorithm with different numbers of particles are run on the same segment of logged data: a 19-min excerpt of a DUC qualifying round at George Air Force Base. The data contains Skynet's sensor measurements of two concentric loops of heavy traffic traveling in opposite directions, recorded as Skynet merged into and out of traffic multiple times. Fig. 14 shows the experimental setup.

The data is presented to variants of the proposed tracking algorithm run with one, two, five, ten, 20, and 50 particles. No truth data is available for the qualifying round, although the merging scenario features repeated instances of the perpendicular intersection encounters and parallel head-on encounters studied in Section IV-A and B. Two new factors are present in this experiment: obstacle occlusion, where one or more obstacles are temporarily blocked from Skynet's view, and large numbers of moving obstacles. This particular data excerpt features large numbers of occlusions occurring primarily when a lead vehicle

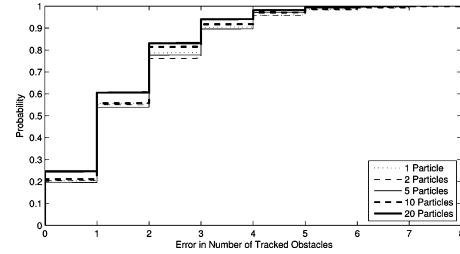


Fig. 15. Sample cumulative distribution function of errors in the number of obstacles tracked in a heavy traffic scenario in variants of the proposed algorithm run with one, two, five, ten, and 20 particles. Errors are calculated against a variant run with 50 particles. The similarities between errors incurred across the wide range of particle configurations demonstrates the system's ability to make data assignments "obvious."

blocks following vehicles from view or when a vehicle in the inner lane occludes one in the outer lane. In this environment, the system tracks an average of 32 potentially moving obstacles at each instant in time.

The close interaction between moving traffic vehicles adds data assignment complexity not present in the first two experiments. Although no truth data is available without tedious and extensive hand tagging efforts, the effects of the added complexity are reflected in the number of obstacles tracked by the system. Incorrect data assignments, even those resolved quickly in resampling, can temporarily result in a particle with too many or too few obstacles. Accurate Monte Carlo sampling of the data assignments, on the other hand, should yield convergence to a common MAP estimate as the number of particles increases. This experiment looks for that convergence across system variants as a measure of data assignment consistency in a complex dynamic environment.

Fig. 15 plots the sample cumulative distribution function of errors in the number of obstacles tracked in variants of the proposed algorithm with one, two, five, ten, and 20 particles. Errors are calculated against a variant run with 50 particles, and are only evaluated for the MAP estimate in each variant, i.e., the particle with the largest weight. Fig. 15 shows that the sample cumulative distribution functions for variants with ten and 20 particles are strictly greater than those of variants with fewer particles, which indicates convergence toward a common number of obstacles as the number of particles increases. More importantly, all the cumulative density functions lie within approximately 5% of each other. From this result, it is evident that the system's point cloud representation and parametric filter representation largely make the data assignments "obvious," as even a single particle is capable of achieving results similar to a 50-particle variant throughout most of the experiment. This performance also indicates the relative stability of the system's errors across a wide variety of practical particle configurations.

Although a single particle performs almost as well as 50 in most of the experiment, there are three isolated instances where variants with low numbers of particles result in significant errors. Two of these instances, occurring at $t \approx 12.235$ and 18.65 min into the data excerpt, correspond to times when Skynet just starts to merge across oncoming traffic. The scenario is difficult

from a data assignment point of view: All traffic vehicles in the oncoming lane instantly become occluded by the closest oncoming vehicle. A metal light pole behind Skynet further compounds the problem, as it temporarily passes in view of Skynet's rear radars at the same point in the maneuver. False positive detections created by a radar speed detector mounted on the light pole could potentially create false obstacles that amplify the discrepancy. The third instance, at $t \approx 7.57$ min, corresponds to Skynet making a 90° left turning around a tight corner. Here again, Skynet's rear radars are likely the source of the error, as a large construction scissors lift temporarily becomes visible during the turn. This lift frequently generates multiple Delphi radar tracks, which may be mistakenly assigned to differing numbers of obstacles across algorithm variants. The problem is further magnified by sparse sensor data, as only rear-facing Delphi radars can see the offending objects. During all three isolated errors, at most eight fewer or eight more objects are tracked than the variant with 50 particles, which tracks 37, 53, and 41 objects in each of the three error periods, respectively. In contrast, Fig. 15 shows that most errors are less than four.

V. PERFORMANCE IN THE DARPA URBAN CHALLENGE

The proposed tracking algorithm has also been implemented on Skynet in real time, where it acts as Skynet's sole obstacle detection and tracking system. The algorithm is implemented in C++ on a single 2.0 GHz Intel Core 2 Duo machine with 2.0 Gb RAM, running Windows Server 2003. Four particles are used in Skynet's implementation to ensure real-time processing. A caching scheme for the measurement function is also implemented to avoid recalculation of $h(o_{m,k}, P_k)$ when obstacle state and point cloud estimates do not change, and assignment likelihoods are only computed for tracked obstacles in thresholded proximity of a measurement. These minor augmentations help to offset the computational expense of numerical differentiation performed in the sigma point transforms utilized in the algorithm.

Skynet relied on the proposed algorithm for obstacle detection and tracking during the 2007 DUC, a 60-mile autonomous urban driving competition held in Victorville, CA, in November 2007. The Urban Challenge featured simultaneous interaction of 11 full-sized autonomous robots and approximately 50 human-driven sedans in typical urban traffic scenarios, such as merging, parking, intersection queuing, and vehicle following. Many of these encounters tested the practical applicability of the system's point cloud representation and stable measurements. During one qualifying round of the Urban Challenge, for example, Skynet was required to complete multiple merges into and out of moving traffic across a lane of oncoming vehicles. Skynet made two attempts at this qualifying course, completing five successful merges into and out of traffic in the first attempt and ten in the second attempt; data from this second attempt was utilized in Section IV-C. In the finals of the Urban Challenge, Skynet was one of only six robots to complete the entire 60-mi course. Skynet typically interacted with one or two other moving traffic vehicles at any given moment in the 60-mile DUC, with a maximum of eight vehicles at once. While competing,

Skynet's system tracked a total of 175 252 distinct obstacles. On average, the system tracked 48.5 obstacles in each particle, with a maximum of 209 obstacles per particle [29].

VI. CONCLUSION

A novel-tracking algorithm has been presented as a computationally feasible, real-time solution to the joint estimation problem of data assignment and dynamic obstacle tracking from a potentially moving robotic platform. The algorithm utilizes an RBPF to separate the joint estimation problem into an independent data assignment problem and a multiple dynamic obstacle-tracking problem conditioned on the data assignments. The RBPF samples the *a posteriori* distribution of data assignments, and compact and efficient parametric filters are used to estimate the *a posteriori* densities of the obstacles conditioned on the sampled assignments. The algorithm achieves a practical computational burden by assigning inexpensive closed-form parametric filters to track large numbers of dynamic obstacles. A novel target representation and stable sensor features have been designed to facilitate successful operation of these filters in the practical application of tracking full-size vehicles from a moving robotic platform. The successes of the proposed target representation and features further reduce the algorithm's dependence on expensive Monte Carlo sampling of data assignment histories.

The algorithm achieves real-time rates through a carefully selected parametric obstacle representation with a data-driven point cloud geometry and stable sensor features. These techniques have been shown to eliminate estimation biases, artifacts, and measurement instability common to sensor data preprocessing techniques, such as box and corner detectors, thereby improving the accuracy of the system's parametric-tracking filters, even under substantial changes in perceived obstacle shape. The result makes data assignments "obvious," and real-time performance is achieved through commensurate reductions in particle requirements in the Monte Carlo approach used to solve the algorithm's data assignment problem.

The algorithm has been implemented on Cornell University's "Skynet," which is an autonomous 2007 Chevrolet Tahoe equipped with a position, velocity and attitude estimator, and laser rangefinders, radars, and optical cameras for obstacle tracking. The algorithm has been validated in three experiments: two experiments in which a single moving target vehicle is tracked under large changes in perceived obstacle shape and one experiment in which multiple moving target vehicles are tracked in heavy traffic. Each of these experiments featured repeated encounters under identical obstacle configurations to achieve statistical significance. In these experiments, the system was shown to be capable of both determining the number of obstacles and accurately tracking their positions and velocities in a statistically unbiased manner. Statistics were also presented for Skynet's performance in the DUC, where the algorithm was implemented in real time to serve as Skynet's obstacle-detection and tracking system. The algorithm ran for 6 h in the Urban Challenge, allowing Skynet to travel autonomously for 60 miles in moving traffic.

Ex.-1019 - ADB Safegate et al.

ADB Safegate et al. v. Honeywell International Inc. – PGR2023-00053

Page 17

REFERENCES

- [1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation: Theory, Algorithms and Software*. New York: Wiley, 2001.
- [2] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. Orlando, FL: Academic, 1988.
- [3] J. Neira and J. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 890–897, Dec. 2001.
- [4] T. Kirubarajan and Y. Bar-Shalom, "Probabilistic data association techniques for target tracking in clutter," *Proc. IEEE*, vol. 92, no. 3, pp. 536–557, Mar. 2004.
- [5] Y. Bar-Shalom, T. Kirubarajan, and X. Lin, "Probabilistic data association techniques for target tracking with applications to sonar, radar, and EO sensors," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 20, no. 8, pp. 37–56, Aug. 2005.
- [6] A. Doucet, V. Ba-Ngu, C. Andrieu, and M. Davy, "Particle filtering for multi-target tracking and sensor management," in *Proc. 5th Int. Conf. Inf. Fusion*, 2002, vol. 1, pp. 474–481.
- [7] S. Särkkä, A. Vehtari, and J. Lampinen, "Rao-blackwellized monte carlo data association for multiple target tracking," in *Proc. 7th Int. Conf. Inf. Fusion*, P. Svensson and J. Schubert, Eds., Mountain View, CA, 2004, vol. 1, pp. 583–590.
- [8] Z. Khan, T. Balch, and F. Dellaert, "MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1960–1972, Dec. 2006.
- [9] L. Lin, Y. Bar-Shalom, and T. Kirubarajan, "Track labeling and PHD filter for multitarget tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 3, pp. 778–795, Jul. 2006.
- [10] R. Mahler, "PHD filters of higher order in target number," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1523–1543, Oct. 2007.
- [11] S. Thrun, "Robotic mapping: A survey," Sch. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-02-111, Feb. 2002.
- [12] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *Int. J. Robot. Res.*, vol. 23, no. 7–8, pp. 693–716, 2004.
- [13] D. Streller, K. Fuerstenberg, and K. Dietmayer, "Vehicle and object models for robust tracking in traffic scenes using laser range images," in *Proc. IEEE 5th Int. Conf. Intell. Transport. Syst.*, Sep. 2002, pp. 118–123.
- [14] K. Fuerstenberg, K. Dietmayer, and S. Eisenlauer, "Multilayer laser scanner for robust object tracking and classification in urban traffic scenes," in *Proc. 9th World Congr. Intell. Transport. Syst.*, Chicago, IL, 2002, pp. 7–16.
- [15] D. Streller and K. Dietmayer, "Object tracking and classification using a multiple hypothesis approach," in *Proc. 2004 IEEE Intell. Veh. Symp.*, Jun., pp. 808–812.
- [16] F. Fayad and V. Cherfaoui, "Tracking objects using a laser scanner in driving situation based on modeling target shape," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2007, pp. 44–49.
- [17] R. MacLachlan and C. Mertz, "Tracking of moving objects from a moving vehicle using a scanning laser rangefinder," in *Proc. IEEE Intell. Transport. Syst. Conf.*, Sep. 2006, pp. 301–306.
- [18] A. Mendes, L. Conde Bento, and U. Nunes, "Multi-target detection and tracking with a laserscanner," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2004, pp. 796–801.
- [19] M. Maehlich, R. Schweiger, W. Ritter, and K. Dietmayer, "Multisensor vehicle tracking with the probability hypothesis density filter," in *Proc. 9th Int. Conf. Inf. Fusion*, 2006, pp. 1–8.
- [20] M. Maehlich, W. Ritter, and K. Dietmayer, "De-cluttering with integrated probabilistic data association for multisensor multitarget acc vehicle tracking," in *Proc. IEEE Intell. Veh. Symp.*, Istanbul, Turkey, 2007, pp. 178–183.
- [21] S. Wender and K. Dietmayer, "3D vehicle detection using a laser scanner and a video camera," *IET Intell. Transp. Syst.*, vol. 2, no. 1, pp. 105–112, Jun. 2008.
- [22] M. Martin and H. Moravec, "Robot evidence grids," Robotics Inst., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-96-06, Mar. 1996.
- [23] I. Miller and M. Campbell, "A mixture-model based algorithm for real-time terrain estimation," *J. Field Robot.*, vol. 23, no. 9, pp. 755–775, 2006.
- [24] C. Wang, "Simultaneous localization, mapping and moving object tracking," Ph.D. dissertation, Robotics Inst., Carnegie Mellon Univ., Pittsburgh, PA, Apr. 2004, Tech. Rep. CMU-RI-TR-04-23.
- [25] J. Nieto, T. Bailey, and E. Nebot, "Scan-SLAM: Combining EKF-SLAM and scan correlation," in *Proc. Field Serv. Robot., Results 5th Int. Conf. Field Serv. Robot., FSR, Port Douglas, QLD, Australia*, New York: Springer-Verlag, 2006, pp. 167–178.
- [26] D. Holz, C. Lörken, and H. Surmann, "Continuous 3D sensing for navigation and SLAM in cluttered and dynamic environments," in *Proc. 11th Int. Conf. Inf. Fusion*, Jun. 2008, pp. 1–7.
- [27] A. Petrovskaya and S. Thrun, "Model based vehicle tracking for autonomous driving in urban environments," in *Proc. Robot.: Sci. Syst. IV*, Jun. 2008.
- [28] L. Fletcher, S. Teller, E. Olson, D. Moore, Y. Kuwata, J. How, J. Leonard, I. Miller, M. Campbell, D. Huttenlocher, A. Nathan, and F.-R. Kline, "The MIT–Cornell collision and why it happened," *J. Field Robot.*, vol. 25, no. 10, pp. 775–807, 2008.
- [29] I. Miller, M. Campbell, D. Huttenlocher, A. Nathan, F.-R. Kline, P. Moran, N. Zych, B. Schimpf, S. Lupashin, E. Garcia, M. Catlin, J. Kurdziel, and H. Fujishima, "Team cornell's skynet: Robust perception and planning in an urban environment," *J. Field Robot.*, vol. 25, no. 8, pp. 493–527, Aug. 2008.
- [30] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, 2008.
- [31] M. Montemerlo *et al.*, "Junior: The stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, 2008.
- [32] A. Bacha *et al.*, "Odin: Team victortango's entry in the DARPA urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 467–492, 2008.
- [33] K. Fuerstenberg, D. Linzmeier, and K. Dietmayer, "Pedestrian recognition and tracking of vehicles using a vehicle based multilayer laserscanner," in *Proc. 10th World Congr. Intell. Transp. Syst.*, Madrid, Spain, 2003.
- [34] J. Leonard *et al.*, "A perception-driven autonomous urban vehicle," *J. Field Robot.*, vol. 25, no. 10, pp. 727–774, 2008.
- [35] J. Bohren *et al.*, "Little ben: The ben franklin racing team's entry in the 2007 DARPA urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 598–614, 2008.
- [36] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using FastSLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sep. 2003, vol. 2, pp. 1985–1991.
- [37] C. Bibby and I. Reid, "Simultaneous localization and mapping in dynamic environments (SLAMIDE) with reversible data association," in *Proc. Robot. Sci. Syst. III*, Atlanta, GA, Jun. 2007.
- [38] F. Dellaert, "Monte carlo EM for data-association and its applications in computer vision," Ph.D. dissertation, Robotics Inst., Carnegie Mellon Univ., Pittsburgh, PA, Sep. 2001.
- [39] Y. Cheng and T. Singh, "Efficient particle filtering for road-constrained target tracking," in *Proc. 8th Int. Conf. Inf. Fusion*, Jul. 2005, vol. 1, pp. 161–168.
- [40] K. Arras, S. Grzonka, M. Luber, and W. Burgard, "Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1710–1715.
- [41] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization mapping problem," in *Proc. AAAI Nat. Conf. Artif. Intell.*, Edmonton, AB, Canada: AAAI, 2002.
- [42] S. Julier and J. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. SPIE: Signal Process., Sens. Fusion, Target Recognit. VI*, 1997, pp. 182–193.
- [43] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River, NJ: Pearson, 2003.
- [44] M. 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.
- [45] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *Proc. 16th Conf. Uncertainty Artif. Intell.*, 2000, pp. 176–183.
- [46] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based slam with Rao-Blackwellized particle filters by adaptive proposals and selective resampling," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 2432–2437.
- [47] I. Miller and M. Campbell, "Rao-blackwellized particle filtering for mapping dynamic environments," in *Proc. Int. Conf. Robot. Autom.*, Apr. 2007, pp. 3862–3869.
- [48] I. Miller, B. Schimpf, J. Leyssens, and M. Campbell, "Tightly-coupled gps/ins system design for autonomous urban navigation," in *Proc. IEEE/ION Posit., Localization, Navigat. Symp.*, 2008, pp. 1297–1310.



Isaac Miller (M'08) received the B.S. degree from the California Institute of Technology, Pasadena, in 2003 and the M.S. and Ph.D. degrees from Cornell University, Ithaca, NY, in 2006 and 2009, respectively, all in mechanical engineering.

He is currently a Senior Research and Development Engineer with Coherent Navigation, Inc., San Mateo, CA. His research interests include robotic perception and planning in the context of Bayesian estimation and probabilistic representation, sensor fusion,

attitude and position estimation, real-time algorithmic implementation for robotic-field deployment, controls, and dynamic modeling.

Dr. Miller was also engaged in the design and implementation of localization and perception algorithms for Cornell University's 2007 Defense Advanced Research Projects Agency (DARPA) Urban Challenge and 2005 DARPA Grand Challenge robots.



Daniel Huttenlocher received the B.Sc. degree from the University of Michigan, Ann Arbor, and the M.Sc. and Ph.D. degrees, both from the Massachusetts Institute of Technology, Cambridge, in 1984 and 1988, respectively.

He is currently the Neafsey Professor of computing, information science, and business with Cornell University, Ithaca, NY, where he holds a joint appointment with the Computer Science Department and the Johnson Graduate School of Management.

He was with Xerox PARC for more than ten years, where he directed work that led to the ISO JBIG2 image-compression standard. His research interests include computer vision, online social networks, autonomous vehicles, and geometric algorithms.

Prof. Huttenlocher is a Fellow of the Association for Computing Machinery. He was the recipient of a National Science Foundation Presidential Young Investigator Award and was named New York State Professor of the Year by the Council for Advancement and Support of Education in 1993.



Mark Campbell (M'00) received the B.S. degree in mechanical engineering from Carnegie Mellon, Pittsburgh, PA, in 1990 and the M.S. and Ph.D. degrees in control and estimation from the Massachusetts Institute of Technology, Cambridge, in 1993 and 1996, respectively.

He is currently an Associate Professor with Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY. His research interests include the estimation and control of autonomous vehicles.

Prof. Campbell was the recipient of a National Aeronautics and Space Administration award for the Middeck Active Control experiment flown on STS-67, Best Paper Awards from the American Institute of Aeronautics and Astronautics (AIAA) and Frontiers in Education, and teaching awards from Cornell, the University of Washington, Seattle, and the American Society of Electrical Engineering (ASEE). He is an Associate Fellow of the AIAA, an Australian Research Council International Fellow. He is the Associate Director of the American Automatic Control Council Board of Directors.