December 2021 Volume 16 Number 2 ISSN 1557-6418

JAIF 21

Journal of Advances in Information Fusion

A semi-annual archival publication of the International Society of Information Fusion

Regular Papers Page

INTERNATIONAL SOCIETY OF INFORMATION FUSION

The International Society of Information Fusion (ISIF) is the premier professional society and global information resource for multidisciplinary approaches for theoretical and applied INFORMATION FUSION technologies. Technical areas of interest include target tracking, detection theory, applications for information fusion methods, image fusion, fusion systems architectures and management issues, classification, learning, data mining, Bayesian and reasoning methods.

Journal of Advances in Information Fusion (ISSN 1557-6418) is published semi-annually by the International Society of Information Fusion. The responsibility for the contents rests upon the authors and not upon ISIF, the Society, or its members. ISIF is a California Nonprofit Public Benefit Corporation at P.O. Box 4631, Mountain View, California 94040.

Data-association-free Characterization of Labeling Uncertainty: the Cross Modeling Tracker

CARLOS MORENO LEON
HANS DRIESSEN
ALEXANDER YAROVOY

The Multiple Object Tracking problem for a known and constant number of closely-spaced objects in a track-before-detect context is considered. The underlying problem of decomposing a data-association-free Bayes posterior density is analyzed. A previously proposed solution for two objects moving in one-dimensional space is generalized for higher dimensional problems where *t* objects move in a M-dimensional space. The underlying problem is solved with the proposed Cross Modeling Tracker by means of hypothesizing physical crosses between the objects for a general *t–MD* objects case. In particular, the mathematical definition of cross-between-objects is generalized from a meaningful interpretation of the problem in the low dimensional setting. A method to provide optimal references for evaluation of the Cross Modeling Tracker is also considered. The Cross Modeling Tracker algorithm is validated with the optimal references by simulating *t–MD* closely-spaced objects scenarios. Wider applicability of the Cross Modeling Tracker with respect to comparable reviewed solutions is demonstrated via simulation experiments.

Manuscript received February 17, 2021; revised July 7, 2021; released for publication December 1, 2021.

Associate Editor: Florian Meyer.

C.M. Leon is with Fraunhofer FHR - Cognitive Radar, Wachtberg, Germany (E-mail: carlos.moreno@fhr.fraunhofer.de).
H. Driessen and A.G. Yarovoy are with Delft University of Technology - Microwave Sensing, Signals and Systems, Delft, The Netherlands (E-mail: {J.N.Driessen, A.Yarovoy}@tudelft.nl)

NOMENCLATURE

MOT	Multiple Object Tracking.
DBT	Detect-before-track.
TBD	Track-before-detect.
RFS	Random Finite Sets.
SNR	Signal to Noise Ratio.
DA	Data-association.
CMT	Cross Modeling Tracker.
MMSE	Minimum Mean Square Error.
JPDA	Joint Probabilistic Data Association.
MHT	Multiple Hypothesis Tracking.
PHD	Probability Hypothesis Density.
LMB	Labeled Multi-Bernoulli.
GOSPA	Generalized Optimal Subpatern Assignment.
PF	Particle Filter.
LPE	Labeled Point Estimates.
MD	M-dimensional.
t	Number of objects.
$\boldsymbol{s}_k$	State vector at time k.
o_k	Variable "order" at time k.
$\boldsymbol{\varkappa}_k$	DBT measurement at time k with indication of correct data association.
$\boldsymbol{Z}_k$	Sequence of DBT measurements up to, and including, time k.
$\boldsymbol{z}_k$	DBT measurement at time k.
$\mathbf{Z}_k$	Sequence of complete TBD measurements up to, and including, time k.
$\mathbf{z}_k$	Complete TBD measurement at time k.
N_r, N_b	Number of range and bearing cells in $\mathbf{z}_k$.
$z_k^{i,j}$	TBD measurement at cell i, j.
N_p	Number of particles.

I. INTRODUCTION

Multiple Object Tracking (MOT) refers to the problem of jointly estimating the presence and states or trajectories of objects based on measurements from sensors such as radars. The majority of reported MOT solutions are only suitable in detect-before-track (DBT) context, i.e., designed for detection measurements. This paper aims to solve the MOT problem for a known and constant number of closely-spaced objects in a track-before-detect (TBD) context.

The problem of deciding which track state estimate belongs to which physical object over time is known as the labeling problem. Labels are considered as unique identifiers assigned to each physical object in the track initiation stage. In closely-spaced objects tracking, sensor systems may not provide enough information to uniquely match objects labels and objects point estimates consistently over time, leading to uncertainty in the labeling [3]. Labeling uncertainty can play a role already in sensor systems even if the objects can be

resolved by the sensor. This will occur if maneuverability of the objects and/or sensor update intervals get large.

The terms labeling problem and labeling uncertainty might be misleading, it is important to remark that there is no uncertainty in the labels as they have been assigned (and are therefore fully known) within the tracker. The uncertainty is about the assignment of tracks to the fully known set of labels [17]. However, due to historical reasons, we will continue referring to this problem and the uncertainty as labeling problem and labeling uncertainty.

Labels have been rigorously incorporated in MOT solutions by two different means. Firstly, by using random vector formulations. In this case, the order of partitions in the random vector implicitly determines the labels of the tracks. Secondly, by using random finite sets (RFS) formations and explicitly introducing labels as additional components in the (unordered) state variable. The scope of this paper is limited to the case of known and constant number of objects in order to isolate the essence of the labeling problem. Under this assumption, a vector formulation suffices.

TBD MOT uses multiple frames of the raw sensor measurements with the objective of avoiding a hard thresholding decision. Consequently, TBD algorithms jointly estimate the existence of the target (detection) as well as track its kinematic state (filtering).

Avoidance of hard thresholding in TBD prevents loss of information, which is remarkably important in the tracking of low signal to noise ratio (SNR) objects. Additionally, TBD MOT cope with closely-spaced objects tracking in cases where a DBT approach would have to deal with merged measurements. Therefore, TBD MOT provides an inherently increased resolution capability over DBT MOT, which however makes it all the more important to address data-association-free characterization of labeling uncertainty.

The advantages of TBD tracking motivate this paper to design an algorithm able to characterize labeling uncertainty without using data-association (DA) techniques. The aim of such algorithm is adding labeling characterization capabilities to regular TBD filters such as the one presented in [6]. The lack of practical solutions is discussed in the Literature Review section. In order to fill the gap, this paper generalizes the association-free framework in [17, Section IV] to arbitrarily high dimensional problems.

The paper is organized as follows. Section II presents the research questions and specifies the underlying problem to answer them. Also, an extensive literature review on the topic is provided. Section III presents a generic formulation to characterize the Cross Modeling Tracker (CMT). Section IV revisits the evaluation method specifically design in [18] to measure estimation performance of the CMT. Sections V and VI provide the analytical generalization of the method presented in [17, Section IV] to the general t–MD objects case. Simulation results are also provided. Section VII draws final conclusions and proposes future research directions.

II. PROBLEM DESCRIPTION AND CONTEXTUALIZATION IN LITERATURE

Consider an MOT problem based on raw (TBD) measurements and a scenario containing t objects moving in a M-dimensional space (t–MD setting). Additionally, consider that t is constant and known by the tracker.

In cases where the objects move far apart from each other, labeling uncertainty is negligible and a regular TBD tracker such as the one in [6] suffices to infer the correct pair-matching between labels and point estimates. However, when the t objects move closely-spaced, labeling uncertainty degrades tracking performance and even the optimal TBD MOT solution is prevented from inferring the correct pair-matching between labels and point estimates. It is important to remark that what we refer to as pair-matching between labels and point estimates is conceptually different from the well-studied pair-matching problem between detections and labels (detections do not even exist in TBD). This difference is described in Appendix A.

Precisely due to the mathematical limitations for providing uncertainty-free labeling in complicated scenarios, estimation of certainty regarding all potential labeling possibilities is a topic of major importance. For this reason, the problem considered in this paper is described by the following two questions:

- What is the list of $t!$ labeled point estimates for the current dynamic state of the objects? Here each labeled point estimate hypothesises the state of the t targets with indication of the labels.
- What is the certainty corresponding to each labeled point estimate in the list?

A regular TBD tracker is known to fail at answering these questions in closely-spaced objects scenarios for two reasons. First, particle-based approximations (required due to the highly non-linear nature of the TBD measurement models) cannot approximate multimodal densities for long periods of time [7]. This can be tackled by using a convenient proposal density [15]. Second, extraction of minimum mean square error (MMSE) point estimates from a TBD multimodal posterior density results in track-coalescence underperformance [5]. This can be solved efficiently by means of characterizing the labeling uncertainty implicitly contained in the TBD multimodal density [17]. The purpose of this paper is providing such characterization for the general t–MD objects case.

A. Formulation of the Problem

The link between the DA problem and the labeling problem is formulated mathematically in Appendix A. It

is concluded that in DBT context, the labeling problem is implicitly solved by tackling the DA problem. Unfortunately, the same analytic method cannot be applied in TBD filtering to solve the labeling problem due to the absence of detections.

In TBD context, $\mathbf{z}_k$ contains the power reflected from the tracking scenario for each radar cell at time step k. For instance, a 2D raw data set $\mathbf{z}_k$ can be modeled as in [6] by $N_r \times N_b$ power measurements z_k^{ij}, where $i = 1, 2, ..., N_r$ and $j = 1, 2, ..., N_b$.

The permutation of measurement $\boldsymbol{z}_k$ described in Appendix A makes no sense for the TBD measurements $\mathbf{z}_k$. Therefore $\pi_m(\mathbf{z}_k)$ (see Appendix A) cannot be used to hypothesize labeling associations in TBD. However, the decomposition using permutations in $\boldsymbol{s}_k$, derived in Appendix A, can be formulated also in TBD:

$$p(\boldsymbol{s}_k|\mathbf{Z}_k) \propto \sum_{m=1}^{t!} l(\mathbf{z}_k|\pi_m(\boldsymbol{s}_k))p(\boldsymbol{s}_k|\mathbf{Z}_{k-1}). \tag{1}$$

We consider the worst case scenario, where TBD measurements do not incorporate information about objects' labels. Under this assumption, the TBD likelihood model for the measurements conditioned on the state $l(\mathbf{z}_k|\boldsymbol{s}_k)$ is invariant with respect to permutation of partitions in the state vector:

$$l(\mathbf{z}_k|\pi_m(\boldsymbol{s}_k)) = l(\mathbf{z}_k|\pi_n(\boldsymbol{s}_k)) \quad \forall\{m, n\} : \{\pi_m, \pi_n\} \in \Pi. \tag{2}$$

This is a remarkable difference with respect to DBT, where $l(\boldsymbol{z}_k|\boldsymbol{s}_k)$ is permutation variant (even when the measurements do not provide any information about the labels). Due to equation (2), equation (1) provides a decomposition in which all components end up being identical disregarding whether objects are closely-spaced or not. For this reason, equation (1) is certainly not relevant for labeling characterization in TBD. In fact, it can be simplified as:

$$\begin{aligned} p(\boldsymbol{s}_k|\mathbf{Z}_k) &\propto \sum_{m=1}^{t!} l(\mathbf{z}_k|\pi_m(\boldsymbol{s}_k))p(\boldsymbol{s}_k|\mathbf{Z}_{k-1}) \\ &\propto l(\mathbf{z}_k|\boldsymbol{s}_k)p(\boldsymbol{s}_k|\mathbf{Z}_{k-1}). \end{aligned} \tag{3}$$

The simplification in equation (3) illustrates that, although $p(\boldsymbol{s}_k|\mathbf{Z}_k)$ can be calculated, an analytical decomposition of $p(\boldsymbol{s}_k|\mathbf{Z}_k)$ cannot be accessed by the TBD filter. This defines the specific problem which needs to be tackled to answer the two questions of interest.

B. Related Literature

A plethora of MOT techniques have been developed over recent decades including Joint Probabilistic Data Association (JPDA) [13], Multiple Hypothesis Tracking (MHT) [21], and Probability Hypothesis Density (PHD) [19]. These algorithms are designed to work with detections. As the scope of this paper is within TBD MOT, none of these methods are suitable.

Some RFS-based trackers use multitrajectory densities instead of multiobject densities [23]. A multitrajectory random variable incorporates the states of the entire history for each trajectory in the set. With this information, track formation is enabled without any type of label incorporation. Although some reported RFS multitrajectory trackers accommodate TBD models [16], answering the two questions in the Problem Description section does not require information about the entire history of the trajectories.

In [14], an RFS-based mutiobject filter is adapted to work in TBD context. The adaptation is based on fitting a particle-based multiobject density approximation as a Labeled Multi-Bernoulli (LMB) RFS density after each filtering iteration. Nonetheless, questions in the Problem Description section cannot be answered. In fact, the so-called Improved LMB filter provides low Generalized Optimal Subpatern Assignment (GOSPA) errors at the expense of losing labeling information.

In [24], a backward simulation method is proposed to recover full trajectory (labeling) information from unlabeled filtering mutiobject densities. However, the process (generating unlabeled filtering posteriors, recovering trajectory information and marginalizing to obtain the filtering density of interest) involves unnecessary complexity overhead. This is specially so in TBD context, where RFS prior conjugacy cannot be exploited as particle-based approximations are required to accommodate highly nonlinear measurement models. Additionally, the problem in the scope of this paper does not even need to be formulated with RFS as the number of objects is assumed known and constant.

Aoki et al. provided a mathematical characterization of the labeling uncertainties with clear physical interpretation in [1]. However, the proposed Multitarget Sequential Monte Carlo filter algorithm involves a computational complexity of $O(N_p^2 t!^2)$. This computational bottleneck is prohibitive for tracking scenarios with more than two objects.

Blom and Bloem [4] introduced a decomposition of the exact Bayes posterior density into the weighted sum of permutation invariant and permutation strictly variant components. The so-called "unique decomposition" was used by García-Fernández in [15] to provide a particle filtering solution relevant for our problem description. The main focus in [15] is on calculating the probability of successful labeling after object separation, where the posterior multiobject density has symmetric nature [8], [20]. In particular, the contribution of García-Fernández to the characterization of labeling uncertainty problem is based on linking the so-called probability of successful labeling to the particular metric and point estimate considered [15]. These contributions where exemplified for the tracking of two objects with wireless sensor networks.

A generalization of the "unique decomposition" [4] was provided by Croise et al. in [9], where the main focus is to demonstrate how this decomposition can be used to

approximate Minimum Mean Optimal Subpartern Assignment (MMOSPA) estimates . Although MMOSPA estimates do not provide answers to the questions in the Problem Description section due to its unlabeled nature, it is important to remark that the consideration of Optimal Subpartern Assignment (OSPA) metric (or its generalization GOSPA) has been reported successful at preventing track-coalescence. However, MMOSPA estimates are very computationally demanding to calculate and therefore its implementation in dynamic systems has only been reported under some approximations [10], [11].

The use of MMOSPA estimates has been encouraged by the majority of reviewed solutions. Only [15] has reported successful results at avoiding track-coalescence using MMSE estimates. This becomes possible as the specific problem formulated in Section II-A is tackled by incorporating characterization of labeling uncertainty in the filtering process. Interestingly, this characterization does not rely on DA techniques and therefore it is usable in TBD. Unfortunately, although proposed as future work in [15], the generalization of this solution for more than two objects has not been addressed.

An alternative solution for incorporating DA-free characterization of labeling uncertainty in the filtering process was provided in [17, Section IV]. Although this solution tackles the problem formulated in Section II-A, it is only usable for the 2–1D objects case. Specifically, the generalization of [17, Section IV] is the gap to be filled with the contribution in this paper.

C. Detailed Review of [17, Section IV]: Modeling Crosses for the 2–1D Objects Case

Fig. 1 illustrates the idea of *cross modeling* in the most simple scenario where the labeling problem appears: two one-dimensional objects approach each other, stay closely-spaced for a while and finally split.

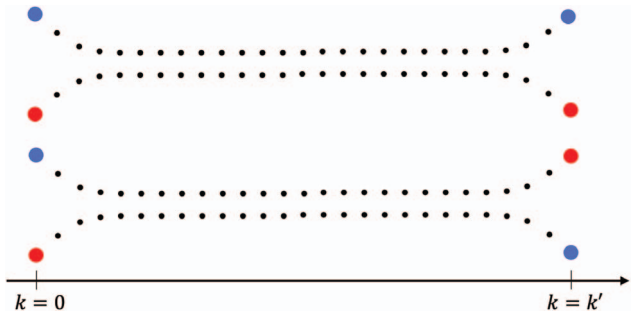


Fig. 1. Representation on top hypothesizes that objects have (physically) crossed an even number of times from $k = 0$ to $k = k'$. Representation at the bottom hypothesizes that objects have (physically) crossed an odd number of times from $k = 0$ to $k = k'$.

At $k = 0$, the tracker places a label to each object according to the tracker's convention: e.g., blue to the target that is further away and red to the other. At any later point in time, for instance $k = k'$, two labeling possibilities are worth considering. One where the furthest object is blue and the nearest one red (example trajectories on top of the figure) and the other with opposite colors, positions being the same (example trajectories at the bottom of the figure).

Note that the questions in Section II do not ask about past states but only current information, for instance at $k = k'$. However, although full trajectory information is not required, it is essential to estimate whether the objects have crossed an even or an odd number of times from $k = 0$ to $k = k'$ in order to characterize the two possible labeling solutions.

Based on described estimation of crosses, [17, Section IV] proposed a method to decompose the association-free TBD multiobject posterior density. For closely-spaced objects situations, the non-decomposed $p(\boldsymbol{s}_k|\mathbf{Z}_k)$ in equation (3) displays symmetric multimodality:

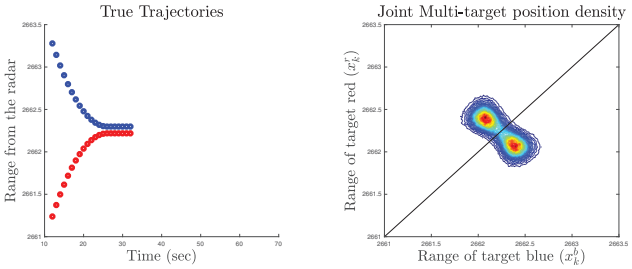


Fig. 2. Illustration of a closely-spaced targets situation. Ground truth trajectories of two targets moving in a line (2–1D objects scenario) in the left-hand side. Last joint multiobject position density (position components of $p(\boldsymbol{s}_k|\mathbf{Z}_k)$) represented in the right-hand side.

One interpretation of the multimodality in $p(\boldsymbol{s}_k|\mathbf{Z}_k)$ with physical meaning is that the objects may have well crossed each other from one time scan to the next one. As one can see in Fig. 2, the probability mass of $p(\boldsymbol{s}_k|\mathbf{Z}_k)$ concentrates in separated regions of the joint state space. These regions represent different labeling permutations of the information of interest.

In [17, Section IV], it is assumed that the state variable is a vector where partitions are stacked: $\boldsymbol{s}_k = [x_k^b\ \dot{x}_k^b\ x_k^r\ \dot{x}_k^r]^T$. Positions of each partition are denoted as x_k and velocities as $\dot{x}_k$. b and r are the labels for the first and second partition respectively (blue and red for printing clarity). In order to hypothesize crosses between objects, the concept of *order* at time step k was defined in [17, Section IV] as:

$$o_k = d(\boldsymbol{s}_k) = \begin{cases} 1 \text{ if } x_k^b > x_k^r \\ 2 \text{ otherwise.} \end{cases} \tag{4}$$

For two objects moving in one dimension, the variable *order* determines whether the position of one 1D object is larger or less than the position of another 1D object. Also, it is trivial to interpret a permutation of *order* in the state vector as a cross of objects between time steps $k-1$ and k:

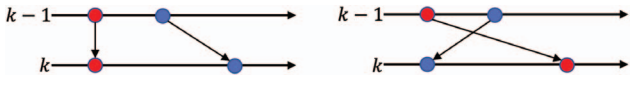


Fig. 3. No permutation of *order* (left-hand side), permutation of *order* (right-hand side).

The notion of order/cross escapes to physical interpretation in higher dimensional problems. One cannot tell whether the position of one 2D object is larger or less than the position of another 2D object as the definition of ">" and "<" is only valid in the 1D line. In fact, although one *order* definition was provided in [17, Section V] for the 2–2D objects case, counterexamples were found proving it wrong.

III. GENERALIZED CHARACTERIZATION OF CROSS MODELING

Fig. 4 illustrates the block diagram of the CMT algorithm required to solve the formulated problem in Section II-A:

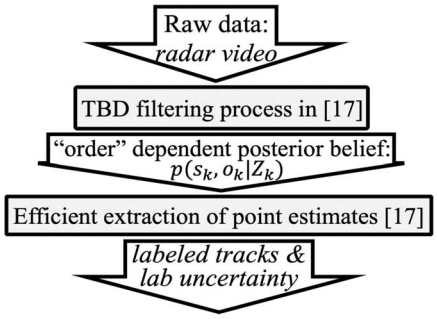


Fig. 4. This TBD framework was pictured in an equivalent way in [17]. Although the block diagram of the algorithm is generic, the definition of o_k in [17, Section IV] does not allow its use out of the 2–1D objects case. Hence, the importance of generalizing the definition of o_k.

The solution to be generalized ([17, Section IV]) was designed for particle-based implementations of the Bayesian filter. In particle filtering, "particle mixing" is a characteristic phenomenon inherent in approximations of multimodal densities [12]. Interestingly for our labeling problem, when "particle mixing" happens at least two particles are represented with permuted order of partitions in the state vector [8]. In order to generalize the definition of order/cross in [17, Section IV] to arbitrarily high dimensional problems, this paper proposes to analyze and exploit "particle mixing" effects in the particle cloud. Exploitation of such "particle mixing" analysis is realized by means of clustering the particle cloud.

Clustering the particle cloud results in remarkable benefits compared to existent methods. These benefits are: straightforward extraction of labeling certainty measures and trouble-free use of computationally efficient MMSE estimation (even when $p(\boldsymbol{s}_k|\mathbf{Z}_k)$ is multimodal).

A. State Space Model

The general nonlinear dynamic system and observation models can be denoted as f and q respectively:

$$\boldsymbol{s}_k = f(\boldsymbol{s}_{k-1}, \boldsymbol{n}_{k-1}), \tag{5}$$

$$\mathbf{z}_k = q(\boldsymbol{s}_k, \boldsymbol{v}_k),\ k \in \mathbb{N}, \tag{6}$$

where $\boldsymbol{s}_k, \boldsymbol{n}_k, \mathbf{z}_k$, and $\boldsymbol{v}_k$ represent the state, process noise, measurement, and measurement noise, respectively.

We tackle the MOT problem in the framework of recursive Bayesian filtering. The predicted and posterior densities of interest $p(\boldsymbol{s}_k|\mathbf{Z}_{k-1})$ and $p(\boldsymbol{s}_k|\mathbf{Z}_k)$ are obtained by iteration over the Chapman–Kolmogórov and Bayes equations:

$$p(\boldsymbol{s}_k|\mathbf{Z}_{k-1}) = \int p(\boldsymbol{s}_k|\boldsymbol{s}_{k-1})p(\boldsymbol{s}_{k-1}|\mathbf{Z}_{k-1})d\boldsymbol{s}_{k-1}, \tag{7}$$

$$p(\boldsymbol{s}_k|\mathbf{Z}_k) = \frac{p(\mathbf{z}_k|\boldsymbol{s}_k)p(\boldsymbol{s}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})}, \tag{8}$$

in this framework, the models in equations (5) and (6) are expressed in the form of $p(\boldsymbol{s}_k|\boldsymbol{s}_{k-1})$ and $p(\mathbf{z}_k|\boldsymbol{s}_k)$, respectively.

Due to the nonlinear nature of TBD measurement models, the Bayesian recursion formulated in equations (7) and (8) cannot be implemented via (stochastic) parametric models [22]. For this reason, a particle filter (PF) will be used to approximate the recursion.

B. A Generic Decomposition of $p(\boldsymbol{s}_k|\mathbf{Z}_k)$ in TBD

As detailed in the Section II-A, the underlying concern to answer the questions in Section II is how to decompose $p(\boldsymbol{s}_k|\mathbf{Z}_k)$ in TBD. A generic decomposition of $p(\boldsymbol{s}_k|\mathbf{Z}_k)$ can be formulated by introducing an auxiliary variable o_k:

$$p(\boldsymbol{s}_k|\mathbf{Z}_k) = \sum_{i=1}^{t!} p(\boldsymbol{s}_k, o_k = i|\mathbf{Z}_k). \tag{9}$$

Please note that the desired definition of o_k is not the one in equation (4) as we are targeting its generalization to the t–MD objects case. The new density of interest $p(\boldsymbol{s}_k, o_k|\mathbf{Z}_k)$, can be factorized as:

$$p(\boldsymbol{s}_k, o_k|\mathbf{Z}_k) = p(\boldsymbol{s}_k|o_k, \mathbf{Z}_k)P(o_k|\mathbf{Z}_k), \tag{10}$$

where $p(\boldsymbol{s}_k|o_k, \mathbf{Z}_k)$ is the posterior density of the state vector given o_k and the measurements, while $P(o_k|\mathbf{Z}_k)$ is the posterior probability of o_k. Both can be computed using the association-free (non-decomposed) TBD filter output $p(\boldsymbol{s}_k|\mathbf{Z}_k)$ and a certain probabilistic definition of the auxiliary variable o_k:

$$p(\boldsymbol{s}_k|o_k, \mathbf{Z}_k) = \frac{P(o_k|\boldsymbol{s}_k, \mathbf{Z}_k)p(\boldsymbol{s}_k|\mathbf{Z}_k)}{P(o_k|\mathbf{Z}_k)}, \tag{11}$$

$$P(o_k|\mathbf{Z}_k) = \int_{\boldsymbol{s}_k} P(o_k|\boldsymbol{s}_k)p(\boldsymbol{s}_k|\mathbf{Z}_k)d\boldsymbol{s}_k. \tag{12}$$

Assuming that the generalized definition of o_k is conditionally independent on $\mathbf{Z}_k$ given $\boldsymbol{s}_k$ (this is the case for the solution in [17, Section IV]), equation (11) can be rewritten as:

$$p(\boldsymbol{s}_k|o_k, \mathbf{Z}_k) = \frac{P(o_k|\boldsymbol{s}_k)p(\boldsymbol{s}_k|\mathbf{Z}_k)}{P(o_k|\mathbf{Z}_k)}. \tag{13}$$

The generic decomposition in equation (9) together with the desired statistical definition of order $P(o_k|\boldsymbol{s}_k)$ should allow answering the questions of interest. When MMSE estimation is the choice for extracting point estimates, the desired definition $P(o_k|\boldsymbol{s}_k)$ should ensure that each component in equation (9) $p(\boldsymbol{s}_k|o_k = m, \mathbf{Z}_k)$ is unimodal. Under this condition, the list of $t!$ label point estimates (question 1) can be provided avoiding track-coalescence. Finally, the desired definition $P(o_k|\boldsymbol{s}_k)$ should be such that $P(o_k = m|\mathbf{Z}_k)$ represent the certainty of labeling association m (question 2).

1) Labeled Point Estimates and Labeling Certainties of the CMT: In a particle-based implementation of the Bayesian recursion, $p(\boldsymbol{s}_k|\mathbf{Z}_k)$ is represented with a weighted set of particles $\{\boldsymbol{s}_k^i, w_k^i\}_{i=1}^{N_p}$. According to the decomposition in equation (9), MMSE order-dependent labeled point estimates (LPEs) can be calculated as the expected value of $\boldsymbol{s}_k$ given the *order* and measurements:

$$\begin{aligned} E[\boldsymbol{s}_k|o_k, \mathbf{Z}_k] &= \int_{\boldsymbol{s}_k} \boldsymbol{s}_k p(\boldsymbol{s}_k|o_k, \mathbf{Z}_k) d\boldsymbol{s}_k \\ &\underset{\text{Eq(13)}}{=} \frac{1}{P(o_k|\mathbf{Z}_k)} \int_{\boldsymbol{s}_k} \boldsymbol{s}_k P(o_k|\boldsymbol{s}_k) p(\boldsymbol{s}_k|\mathbf{Z}_k) d\boldsymbol{s}_k \\ &\approx \frac{1}{P(o_k|\mathbf{Z}_k)} \sum_i w_k^i \boldsymbol{s}_k^i P(o_k|\boldsymbol{s}_k^i). \end{aligned} \tag{14}$$

Order-dependent labeling certainties are necessarily in the second factor of equation (10):

$$P(o_k|\mathbf{Z}_k) = \int_{\boldsymbol{s}_k} P(o_k|\boldsymbol{s}_k) p(\boldsymbol{s}_k|\mathbf{Z}_k) d\boldsymbol{s}_k \approx \sum_i w_k^i P(o_k|\boldsymbol{s}_k^i). \tag{15}$$

Evaluation of $E[\boldsymbol{s}_k|o_k, \mathbf{Z}_k]$ for each possible realization of o_k provides a different vector of LPEs (answer to question 1):

$$\begin{aligned} E[\boldsymbol{s}_k|o_k = m, \mathbf{Z}_k] &\approx \frac{1}{P(o_k = m|\mathbf{Z}_k)} \sum_i w_k^i \boldsymbol{s}_k^i P(o_k = m|\boldsymbol{s}_k^i) \\ &\approx \frac{1}{P(o_k = m|\mathbf{Z}_k)} \sum_{i|o_k^i = m} w_k^i \boldsymbol{s}_k^i \quad \text{where } m \in \{1, 2, .., t!\}. \end{aligned} \tag{16}$$

In the same way, evaluation of $P(o_k|\mathbf{Z}_k)$ for each possible realization of o_k provides a scalar with the associated labeling probability (answer to question 2):

$$P(o_k = m|\mathbf{Z}_k) \approx \sum_i w_k^i P(o_k = m|\boldsymbol{s}_k^i) \approx \sum_{i|o_k^i = m} w_k^i. \tag{17}$$

2) Algorithm Implementation: Alg. 1 is the practical implementation of the functionalities illustrated in Fig. 4.

ALGORITHM 1 Pseudo-code of the PF algorithm for implementation of the CMT. Extensions over the plain SIR TBD particle filter plus traditional MMSE estimate extraction are highlighted in blue color. $p(\boldsymbol{s}_k)$ and $p(\boldsymbol{n}_k)$ denote the initial prior and the process noise models.

1 $k = 0$
2 Draw N_p samples $\boldsymbol{s}_k^i$ from $p(\boldsymbol{s}_k)$
3 Draw N_p samples $\boldsymbol{n}_k^i$ from $p(\boldsymbol{n}_k)$
4 $k = k + 1$
5 $\boldsymbol{s}_k^i = f(\boldsymbol{s}_{k-1}^i, \boldsymbol{n}_{k-1}^i)$
6 Calculate o_k^i according to the definition of "order" under test
7 Given $\mathbf{z}_k$, obtain $\tilde{w}_k^i = p(\mathbf{z}_k|\boldsymbol{s}_k^i)$
8 Normalize weights $w_k^i = \tilde{w}_k^i / \sum_{j=1}^{N_p} \tilde{w}_k^j$
9 Resample from $\hat{p}(\boldsymbol{s}_k|\mathbf{Z}_k) = \sum_{i=1}^{N_p} w_k^i \delta(\boldsymbol{s}_k - \boldsymbol{s}_k^i)$:
10 Extract LPEs according to equation (16)
11 Obtain certainty measures of LPEs according to equation (17)
go to 3

IV. PERFORMANCE EVALUATION OF THE CROSS MODELING (CM) METHOD

Before delving into the main contribution of this paper, this section provides an optimal reference for future evaluation of the generalized CMT. Also, the validation of the solution in [17, Section IV] is provided in this section. Two algorithms implementing the optimal reference were provided in [18] based on the link between the DA problem and the labeling problem. In order to provide a self-contained explanation, the essential theory required to build up the optimal references is provided in Appendix A. The method we apply to validate the CMT can be summarized as follows:

In DBT context and taking the considerations from the sixth bullet point in Appendix A.

- Generate the analytical decomposition of $p(\boldsymbol{s}_k|\mathbf{Z}_k)$:

$$p(\boldsymbol{s}_k|\mathbf{Z}_k) \propto \sum_{m=1}^{t!} l(\pi_m(\mathbf{z}_k)|\boldsymbol{s}_k) p(\boldsymbol{s}_k|\mathbf{Z}_{k-1}). \tag{18}$$

- For each data-association-dependent component $l(\pi_m(\mathbf{z}_k)|\boldsymbol{s}_k)p(\boldsymbol{s}_k|\mathbf{Z}_{k-1})$
  - Extract MMSE labeled point estimates (optimal reference):

$$\begin{aligned} &E[p(\pi_m(\mathbf{z}_k)|\boldsymbol{s}_k)p(\boldsymbol{s}_k|\mathbf{Z}_{k-1})] \\ &= \int_{\boldsymbol{s}_k} \boldsymbol{s}_k p(\pi_m(\mathbf{z}_k)|\boldsymbol{s}_k) p(\boldsymbol{s}_k|\mathbf{Z}_{k-1}) d\boldsymbol{s}_k. \end{aligned} \tag{19}$$

– Extract labeling certainty associated (optimal reference):

$$p(\pi_m(\boldsymbol{z}_k)|\boldsymbol{Z}_k) = \frac{\int l(\pi_m(\boldsymbol{z}_k)|\boldsymbol{s}_k)p(\boldsymbol{s}_k|\boldsymbol{Z}_{k-1})d\boldsymbol{s}_k}{\sum_{m=1}^{t!}\int l(\pi_m(\boldsymbol{z}_k)|\boldsymbol{s}_k)p(\boldsymbol{s}_k|\boldsymbol{Z}_{k-1})d\boldsymbol{s}_k}. \tag{20}$$

- Decompose $p(\boldsymbol{s}_k|\boldsymbol{Z}_k)$ from equation (18) using the CMT decomposition (equation (9)):

$$p(\boldsymbol{s}_k|\boldsymbol{Z}_k) = \sum_{m=1}^{t!} p(\boldsymbol{s}_k, o_k = m|\boldsymbol{Z}_k). \tag{21}$$

- For each data-association-free component $p(\boldsymbol{s}_k, o_k = m|\boldsymbol{Z}_k)$, factorize it as in equation (10) and:
  – Extract CMT MMSE labeled point estimates according to equation (16).
  – Extract CMT labeling certainty associated according to equation (17).

It is very important to remark that, although the CMT is motivated by the needs of TBD filtering, performance of the CMT can only be evaluated in the context of DBT. This becomes apparent when considering the evaluation method described above this lines. Please note that the optimal references in equations (19) and (20) need to be derived from the DBT analytical expression in equation (18). Nonetheless, the validation in DBT guaranties equivalent estimation performance of the CMT in TBD. This is because the CMT is designed to infer the relevant DA-free decomposition of any posterior density, disregarding whether it (the posterior density) was generated in the context of DBT or TBD.

Estimation performance of the CMT can be measured as the quality of the decomposition in equation (21). This is done in following sections by using the results in equations (19) and (20) as the optimal reference to answer the questions in Section II. For the validation of the proposed generalization of "order," the results in equations (16) and (17) will be compared with the optimal reference.

A. Algorithm Implementation

Alg. 2 implements the generation of the optimal references to answer questions in Section II based on the decomposition of equation (18). Concerning the second Remark in the Appendix A, $p(\boldsymbol{s}_k|\boldsymbol{Z}_{k-1})$ is commonly formulated as a mixture of densities in algorithms such as MHT, leading to exponentially increasing number of hypotheses over time. However, in the context of this paper, equation (18) is implemented with a single particle filter. Therefore, $p(\boldsymbol{s}_k|\boldsymbol{Z}_{k-1})$ can be formulated as a single density, even when it has multimodal nature.

ALGORITHM 2 Pseudo-code for generation of optimal references (lines 9 and 10) to answer questions in Section II.

1 $k = 0$
2 Draw N_p samples $\boldsymbol{s}_k^j$ from $p(\boldsymbol{s}_k)$
3 Draw N_p samples $\boldsymbol{n}_k^j$ from $p(\boldsymbol{n}_k)$
4 $k = k + 1$
5 $\boldsymbol{s}_k^j = f(\boldsymbol{s}_{k-1}^j, \boldsymbol{n}_{k-1}^j)$
Generate data-association-dependent posterior beliefs
6 for i=1 until t!
7 Given $\boldsymbol{z}_k$, obtain $\tilde{w}_k^{i,j} = p(\pi_i(\boldsymbol{z}_k)|\boldsymbol{s}_k^j)$
8 Normalize weights $w_k^{i,j} = \tilde{w}_k^{i,j} / \sum_{j=1}^{N_p} \tilde{w}_k^{i,j}$
9 Extract LPEs according to the particle – based approximation of equation (19)
10 Obtain certainty measures of LPEs according to particle – based approximation of equation (20)
Sum data-association-dependent posteriors in order to approximate $p(\boldsymbol{s}_k|\boldsymbol{Z}_k)$ with a single particle cloud
11 for j=1 until N_p
12 $w_k'^j = \sum_{i=1}^{t!} w_k^{i,j}$
13 Normalize weights $w_k^j = w_k'^j / \sum_{j=1}^{N_p} w_k'^j$
14 Resample N_p times from $\hat{p}(\boldsymbol{s}_k|\boldsymbol{Z}_k) = \sum_{j=1}^{N_p} w_k^j \delta(\boldsymbol{s}_k - \boldsymbol{s}_k^j)$ to generate $\hat{p}(\boldsymbol{s}_k|\boldsymbol{Z}_k) = \frac{1}{N_p}\sum_{j=1}^{N_p} \delta(\boldsymbol{s}_k - \boldsymbol{s}_k^j)$
go to 3

B. Performance Evaluation of [17, Section IV]

The choice of simulation parameters to evaluate the solution in [17, Section IV] is shown in Table I. This parametrization generates high amount of labeling uncertainty in order to challenge the MOT algorithm. This becomes apparent when considering the overlapping in between partitions of different colors (labels) of the predicted and posterior particle cloud in Fig. 5.

The evaluation can be reproduced by setting up the trajectories for the objects with initial and minimum distance as indicated by d_i and d_m in Table I. The evaluation considers the dynamic and measurement models as nearly constant velocity [2] and linear-Gaussian. The standard deviations of the process noise and observation noise are indicated by σ_n and σ_v, respectively, in Table I,

TABLE I
Parameters of the simulation

Parameter	Value
d_i	1.66 m
d_m	0.22 m
σ_n	0.14 m/s$^{3/2}$
σ_v	0.045 m
τ	1 s
N_p	10000 particles

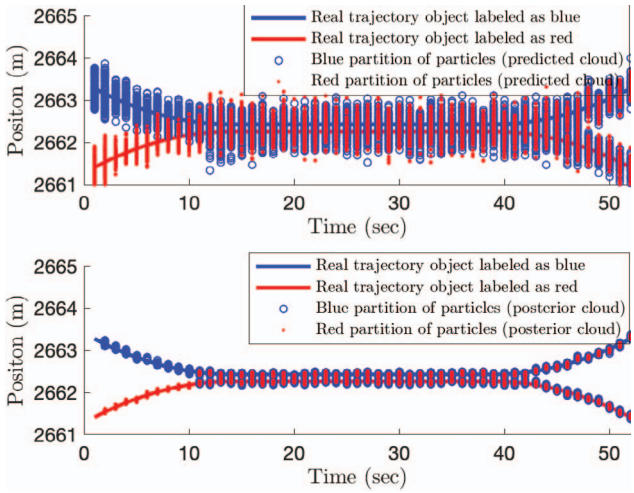


Fig. 5. Illustration of real trajectories as well as predicted and posterior particle clouds along all simulation time steps. The particle mixing effect can be observed, for instance, in the posterior particle clouds after object separation. In fact, different particles hypothesize the state of the same physical object with different partitions.

where τ denotes the revisit time. Without loss of generality, choosing these simple models suffices to validate Alg. 1 with Alg. 2.

The definition of o_k in equation (4) can be rewritten in a probabilistic form:

$$\begin{aligned} P(o_k = 1|\boldsymbol{s}_k) &= 1 \;\; if \;\; x_k^b > x_k^r, \\ P(o_k = 2|\boldsymbol{s}_k) &= 1 \;\; if \;\; x_k^b \leq x_k^r. \end{aligned} \tag{22}$$

The definition in equation (22) can be plugged into equations (13) and (12) in order to decompose $p(\boldsymbol{s}_k|\boldsymbol{Z}_k)$ using the CMT (Alg. 1). By these means, LPEs and associated labeling certainties can be generated with Alg. 1 and evaluated with the optimal reference generated by Alg. 2. Simulation results illustrated in Figs. 6 and 7 validate the cross modeling solution for the 2–1D objects case using the evaluation method described along this section.

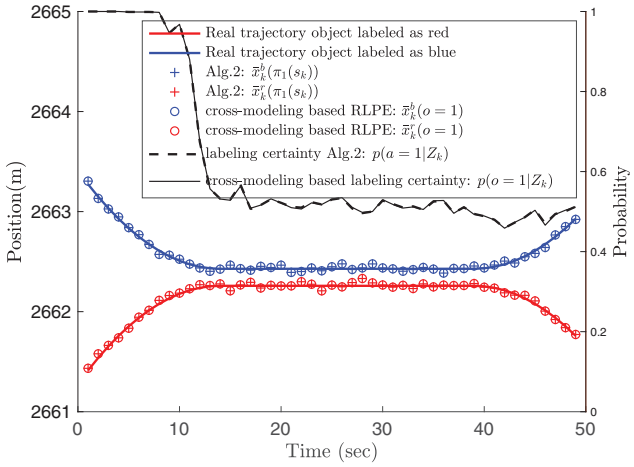


Fig. 6. Evaluation of estimation performance of the CM method for a 2–1D objects scenario. Cross modeling based estimates are extracted only from the cluster associated to "order" 1.

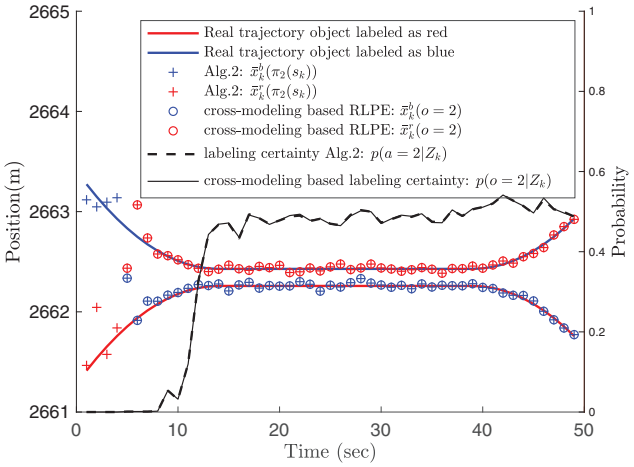


Fig. 7. Evaluation of estimation performance of the CM method for a 2–1D objects scenario. Cross modeling based estimates are extracted only from the cluster associated to "order" 2.

1) Comments on the Results: Alg. 1 decomposes the particle cloud in two ($t!$) clusters. These clusters explicitly approximate the DA-free decomposition in equation (9). The cluster associated to "order" 1 supports the hypothesis that the maneuvers of the two labeled objects lead to non-crossed trajectories with respect to the initialization $\forall k$. This cluster produces LPEs represented in Fig. 6 as $\bar{x}_k^{b,r}(o=1)$. The certainty measure associated ($P(o=1|\boldsymbol{Z}_k)$ curve) is extracted from the particle approximation of the joint multiobject posterior in step 11 of Alg. 1.

Over the first 4 seconds of the simulation, the LPE($o=1$) holds full certainty. No single particle belongs to the cluster $o=2$ and therefore, no representative can be extracted from there (see Fig. 7). As expected, $P(o_k=1|\boldsymbol{Z}_k)$ drops down to around 0.5 after the objects remain closely-spaced for some time. This means that labeling information has been completely lost. Labeling information cannot be recovered after the split as suggested by the measure of certainty. The LPE($o=2$) supports the hypothesis that the maneuvers of the two labeled objects lead to crossed trajectories with respect to the initialization $\forall k$.

Estimation results produced by the CMT closely match the optimal references generated by Alg. 2. This validates the CM method as a solution to the problem defined in the Section II-A for the 2–1D objects scenario.

V. FIRST GENERALIZATION OF THE DEFINITION OF *cross*: SCENARIOS WITH ARBITRARY NUMBER t OF 1D OBJECTS

For 2–1D object settings, equation (4) provides the *order* of any particle given its state. We will refer to this calculation as an absolute evaluation of *order*. Unlike 1-D points, 2-D points cannot be ordered making use of

the operators $\leq$ and $>$. For this reason, finding an absolute evaluation of "order" in 2-D is not a trivial concern.

A. Absolute Versus Relative *Order* Calculation and the Definition of *Cross*:

An alternative method to obtain the *order* can be realized by relative evaluation: relying on the prior *order* at time step $k-1$ and detecting whether or not the objects have crossed from time step $k-1$ to k,

	$P(o_k = 1\|\boldsymbol{s}_k) = 1$	$P(o_k = 2\|\boldsymbol{s}_k) = 1$
$P(o_{k-1} = 1\|\boldsymbol{s}_{k-1}) = 1$	no cross	cross
$P(o_{k-1} = 2\|\boldsymbol{s}_{k-1}) = 1$	cross	no cross

For 2–1D objects scenarios:

	$x_k^r < x_k^b$	$x_k^r \geq x_k^b$
$x_{k-1}^r < x_{k-1}^b$	no cross	cross
$x_{k-1}^r \geq x_{k-1}^b$	cross	no cross

Therefore, for a cross to be declared, one of the two following conditions should hold:

$$x_{k-r}^r < x_{k-r}^b \quad and \quad x_k^r \geq x_k^b, \tag{23}$$

$$x_{k-r}^r \geq x_{k-r}^b \quad and \quad x_k^r < x_k^b. \tag{24}$$

The relative "order" evaluation method shifts the generalization problem from the definition of *order* to the definition of *cross*. A *cross* detector for 2–1D objects can be derived from the absolute definition in equation (4). Let us denote $\boldsymbol{s}_k'^p$ as the position part of the state vector $\boldsymbol{s}_k = [x_k^b\ \dot{x}_k^b\ x_k^r\ \dot{x}_k^r]$ of particle p at time step k: $\boldsymbol{s}_k'^p = [x_k^{p,b}\ x_k^{p,r}]$. Since $\boldsymbol{s}_k'^p$ and $\boldsymbol{s}_{k-1}'^p$ are vectors in a 2D space (two objects placed along 1D spacial dimension), one can calculate the Euclidean distance or $l2-norm$ between them in the joint space as (superscript p is dropped for notation simplicity):

$$norm(\boldsymbol{s}_k' - \boldsymbol{s}_{k-1}') = \sqrt{(x_k^r - x_{k-1}^r)^2 + (x_k^b - x_{k-1}^b)^2} \tag{25}$$

$$= \sqrt{(x_k^r)^2 - 2(x_k^r x_{k-1}^r) + (x_{k-1}^r)^2 + (x_k^b)^2 - 2(x_k^b x_{k-1}^b) + (x_{k-1}^b)^2} \tag{26}$$

$$= \sqrt{(x_k^b - x_{k-1}^r)^2 + (x_k^r - x_{k-1}^b)^2 + 2(x_{k-1}^r - x_{k-1}^b)(x_k^b - x_k^r)} \tag{27}$$

Then,

$$norm^2(\boldsymbol{s}_k' - \boldsymbol{s}_{k-1}') = (x_k^b - x_{k-1}^r)^2 + (x_k^r - x_{k-1}^b)^2 + 2(x_{k-1}^r - x_{k-1}^b)(x_k^b - x_k^r) \tag{28}$$

$$norm^2(\boldsymbol{s}_k' - \boldsymbol{s}_{k-1}') = norm^2(\Pi\boldsymbol{s}_k' - \boldsymbol{s}_{k-1}') + K \tag{29}$$

where $K = 2(x_{k-1}^r - x_{k-1}^b)(x_k^b - x_k^r)$ and Π denotes the permutation matrix:

$$\Pi = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{30}$$

Each of the equations (23) and (24) define conditions for a cross to be declared. Interestingly, when any of these equations is applied to function K, the result is $K > 0$ as long as $x_{k-1}^r, x_{k-1}^b, x_k^r$ and x_k^b take positive values. Variables $x_{k-1}^r, x_{k-1}^b, x_k^r$, and x_k^b can only take positive values in our application problem as these are range magnitudes. Therefore, inequation (31) holds as long as a cross in the state vector of particle p takes place:

$$norm^2(\boldsymbol{s}_k'^p - \boldsymbol{s}_{k-1}'^p) \ > \ norm^2(\Pi\boldsymbol{s}_k'^p - \boldsymbol{s}_{k-1}'^p). \tag{31}$$

Equivalently, the next *order switch* condition can be used to find out whether or not a cross should be declared for the particle p between time instants $k-1$ and k.

$$norm(\boldsymbol{s}_k'^p - \boldsymbol{s}_{k-1}'^p) \ > \ norm(\Pi\boldsymbol{s}_k'^p - \boldsymbol{s}_{k-1}'^p). \tag{32}$$

The result in equation (32) can be pictured in a physically meaningful way:

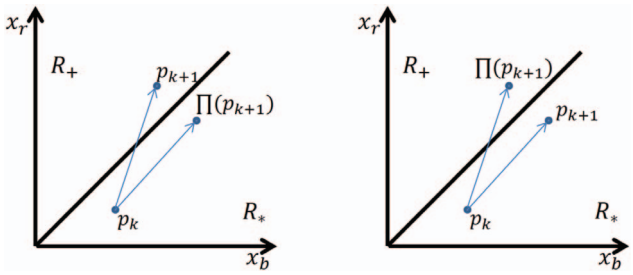


Fig. 8. The illustration in the left hand side represents an *order switch* as the points p_k and p_{k+1} belong to different sides of the diagonal $x_r = x_b$. Indeed, inequation (32) holds in this case. In the right hand side the point p does not cross the line $x_r = x_b$ between time instants k and $k+1$ meaning that the order remains the same in this case. Indeed, inequation (32) does not hold.

B. Generalization of *Cross* from 2–1D to t–1D Cases

This generalization considering 1D objects makes use of the relative definition of "order" based on inequation (32), applied to all possible pairs of partitions in the state vector. As the number of possible labeling solutions is $t!$, only the extension from two to three targets will be exemplified due to space limitations. Nevertheless, there is no loss of generality as the same technique can be applied to any arbitrary number of targets t.

For a three-object scenario, the three partitions in the state vector can be ordered in six different ways according to, for instance, the following convention:

TABLE II
As an example, this convention will group in cluster 1 those particles complying with: position of the first partition is less than the position of the second partition, being the position of the second partition less than the position of the third partition. For the sake of printing clarity in forthcoming simulation experiments, the first, second and third partitions will be identified with colors green, red, and blue, respectively.

	1D magnitude increasing direction →
$o=1$	$1^{st}\ 2^{nd}\ 3^{rd}$
$o=2$	$1^{st}\ 3^{rd}\ 2^{nd}$
$o=3$	$2^{nd}\ 1^{st}\ 3^{rd}$
$o=4$	$3^{rd}\ 1^{st}\ 2^{nd}$
$o=5$	$2^{nd}\ 3^{rd}\ 1^{st}$
$o=6$	$3^{rd}\ 2^{nd}\ 1^{st}$

Between time steps $k-1$ and k, there exist eight different types of crosses which could happen. Each cross type is composed of three boolean variables. These are used to codify whether or not a cross is declared between pairs of partitions. When the boolean variable is set to 1, a cross is declared between the corresponding pair of objects. For instance, one can adopt the following convention:

	Cross of partitions $1^{st}-2^{nd}$	Cross of partitions $1^{st}-3^{rd}$	Cross of partitions $2^{nd}-3^{rd}$
C1	0	0	0
C2	0	0	1
C3	1	0	0
C4	0	1	1
C5	1	1	0
C6	1	1	1
C7	1	0	1
C8	0	1	0

These cross-type codes can be used now for relative evaluation of the *order* at time step k, given the order at time step $k-1$.

	$o_k=1$	$o_k=2$	$o_k=3$	$o_k=4$	$o_k=5$	$o_k=6$
$o_{k-1}=1$	C1	C2	C3	C4	C5	C6
$o_{k-1}=2$	C2	C1	C7	C8	C6	C5
$o_{k-1}=3$	C3	C7	C1	C6	C8	C4
$o_{k-1}=4$	C4	C8	C6	C1	C7	C3
$o_{k-1}=5$	C5	C6	C8	C7	C1	C2
$o_{k-1}=6$	C6	C5	C4	C3	C2	C1

C. Simulation Results

The estimation performance of the CM method for a 3–1D objects scenario, using inequation (32) as the "cross detector", is illustrated in this subsection. Note that the validation method summarized in Section IV is based on evaluation of data association hypotheses. Therefore, Alg. 2 can be used right away to generate optimal LPEs and labeling probabilities in any arbitrary t–MD objects case. Fig. 9 illustrates the estimation performance of the CM tracker.

1) Discussion of Results: The results in Fig. 9 reveal remarkably accurate estimation performance of the CMT both in the computation of LPEs and labeling probabilities. This validates the CMT as a convenient solution to answer the questions in Section II for scenarios where an arbitrary number of objects move in one dimension.

The choice of the particular ground truth trajectories in Fig. 9 results in complete loss of labeling information (1/3! certainty for all LPEs). Although Fig. 9 illustrates this worst case scenario, it is apparent that the CMT accurately estimates labeling uncertain also in more favorable scenarios. For instance, scenarios where the objects remain closely-spaced for a shorter time and labeling certainty is lost only partially.

Disregarding the scenario, losing labeling information is a physical limitation inherent in closely-spaced object scenarios. Furthermore, once labeling certainty is lost, is mathematically not possible to recover it (assuming that the objects cannot be differentiated in measurements/maneuverability). In this context, all what can be expected from the tracker is that it captures the uncertainty produced by the combination scenario/sensor-limitations as accurately as possible. While regular TBD filters such as the one in [6] fail at doing so, the CMT reports successful results for the cases simulated so far.

VI. SECOND GENERALIZATION OF THE DEFINITION OF *cross*: FROM t–1D TO t–MD OBJECTS

Section III-B presented a generic DA-free decomposition of $p(\boldsymbol{s}_k|\mathbf{Z}_k)$ by introducing the (not yet defined) variable o_k. An analytical expression for the LPEs and labeling probabilities (dependent on the definition of o_k) was provided in equations (16) and (17). Another analytical derivation provided in Section V-A has proved that the relative calculation of o_k for the 2–1D objects case, based on inequation (32), leads to identical results than the absolute calculation of o_k based on equation (4).

Section V-B illustrated how the relative calculation of o_k [based on inequation (32)] can be extended seamlessly to cope with more than two objects moving in a one-dimensional space. This section covers the generalization of the relative calculation of o_k from 1D to MD objects. In practice, simulation experiments will be limited to cases where M is 2 and 3 to illustrate that the CMT

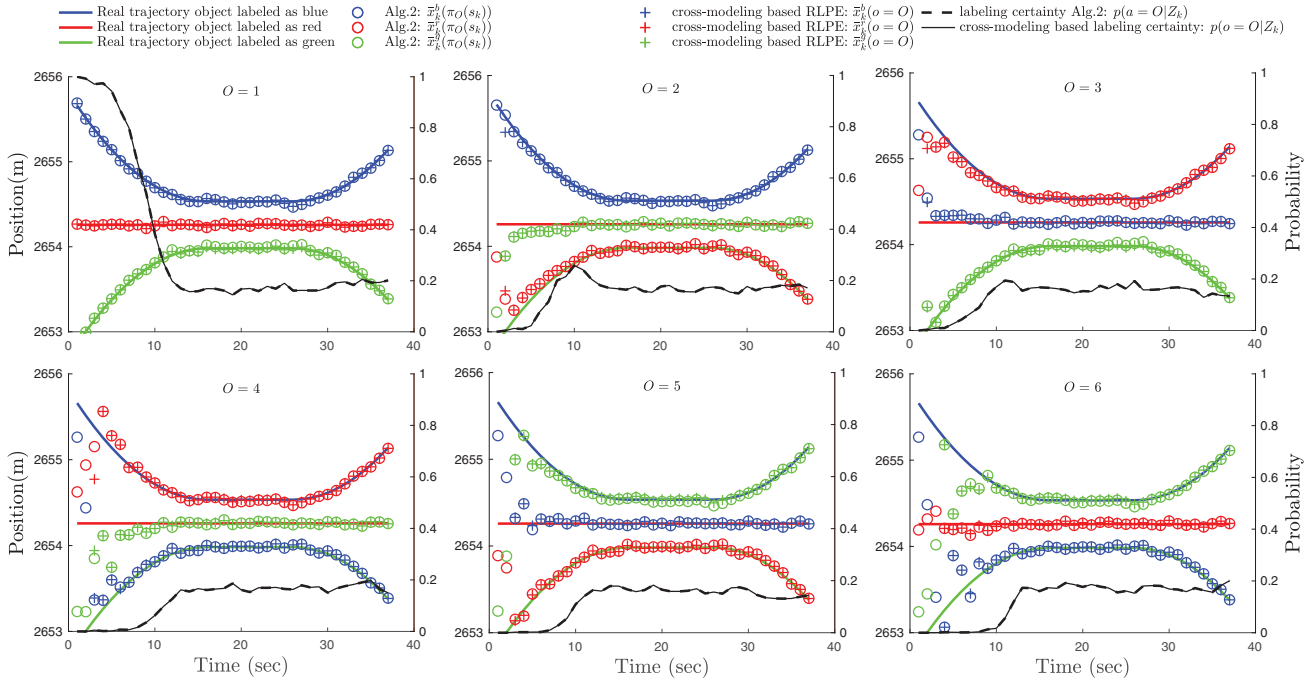


Fig. 9. Evaluation of estimation results provided from the clusters associated to all states of "order" in Table II.

can be used for tracking land/sea objects (2D) and air objects (3D).

The reader is referred to Appendix B at this point in order to understand the complexity of extending the definition of o_k in equation (4) to objects moving in 2 and 3 spacial dimensions.

A. Generalization of the *Cross* Detector for 2–2*D* Objects Settings

Inequation (32) does not extend straightforwardly from 2–1*D* to 2–2*D* objects scenarios as illustrated in Appendix B. However, the reason why inequation (32) works well for the 2–1*D* case can be found after analyzing the function K derived in Section V-A. Given the 2D points $\boldsymbol{s}'^p_{k-1}$ and $\boldsymbol{s}'^p_k$ (two objects in a 1D space), the function $K = 2(x^r_{k-1} - x^b_{k-1})(x^b_k - x^r_k)$ complies with a very particular condition:

$$K(\boldsymbol{s}'^p_{k-1}, \boldsymbol{s}'^p_k) = -K(\boldsymbol{s}'^p_{k-1}, \Pi(\boldsymbol{s}'^p_k)), \tag{33}$$

as $\boldsymbol{s}'^p_k$ and $\Pi(\boldsymbol{s}'^p_k)$ are crossed with respect to each other, K can be regarded as an odd function in the "order" of $\boldsymbol{s}'^p_k$. In other words, two different evaluations of K, using the current state of particle p and its permuted version are equal in absolute value but different in sign:

$$\begin{aligned} |K(\boldsymbol{s}'^p_{k-1}, \boldsymbol{s}'^p_k)| &= |K(\boldsymbol{s}'^p_{k-1}, \Pi(\boldsymbol{s}'^p_k))| \\ sgn(K(\boldsymbol{s}'^p_{k-1}, \boldsymbol{s}'^p_k)) &= sgn(-K(\boldsymbol{s}'^p_{k-1}, \Pi(\boldsymbol{s}'^p_k))). \end{aligned} \tag{34}$$

Additionally, as we already pointed out in Section V-A:

$$K(\boldsymbol{s}'^p_{k-1}, \boldsymbol{s}'^p_k) > 0 \quad \Leftrightarrow \quad \boldsymbol{s}'^p_{k-1}, \boldsymbol{s}'^p_k \textit{ are crossed}. \tag{35}$$

The condition $K(\boldsymbol{s}'^p_{k-1}, \boldsymbol{s}'^p_k) > 0$ and inequation (32) are equivalent "cross detectors" for the 2–1*D* case due to equation (29). Our problem formulation can be narrowed down to the following question: What is the 2–2*D* counterpart of the K function which can be applied to the two 4D points $\boldsymbol{s}'^p_{k-1}$ and $\boldsymbol{s}'^p_k$?

Let us consider an equivalent expression of K introducing the *norm* function, which we denote as K_{2-1D}:

$$\begin{aligned} K = K_{2-1D} = 2(norm(\boldsymbol{s}'^r_{k-1}) - norm(\boldsymbol{s}'^b_{k-1}))(norm(\boldsymbol{s}'^b_k) \\ -norm(\boldsymbol{s}'^r_k)). \end{aligned} \tag{36}$$

The desired function $K_{2-2D} : \mathbb{R}^8 \rightarrow \mathbb{R}$ can be found by considering the counterpart of K_{2-1D} for the 2–2*D* case:

$$\begin{aligned} K_{2-2D} = 2(norm(\boldsymbol{s}'^r_{k-1}) - norm(\boldsymbol{s}'^b_{k-1}))(norm(\boldsymbol{s}'^b_k) \\ -norm(\boldsymbol{s}'^r_k)). \end{aligned} \tag{37}$$

where now $\boldsymbol{s}'^p_k = [x^{p,b}_k\ y^{p,b}_k\ x^{p,r}_k\ y^{p,r}_k]^T$, $\boldsymbol{s}'^{p,r}_k = [x^{p,r}_k\ y^{p,r}_k]^T$ and $\boldsymbol{s}'^{p,b}_k = [x^{p,b}_k\ y^{p,b}_k]^T$. K_{2-2D} is indeed the odd function (in the "order" of $\boldsymbol{s}'_k$) which complies with the conditions in equations (33) and (35). Therefore, the extension of the definition of "cross" for the 2–2*D* objects case is:

$$K_{2-2D} > 0. \tag{38}$$

1) Simulations Results: The results provided by the CMT for the 2–2*D* objects case using inequation (38) as the "cross detector" are shown in Fig. 10.

B. Discussion of the Results

The results reveal remarkably accurate estimation performance of the CMT both in the computation of LPEs and labeling probabilities. This validates the CMT as a convenient solution to answer the questions in Section II for 2D objects. In fact, LPEs result in low

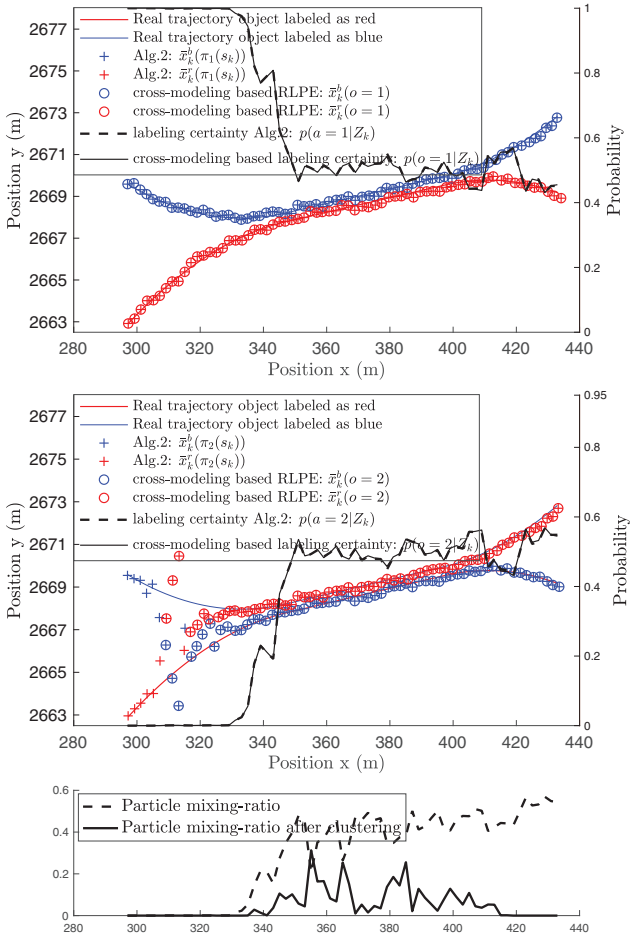


Fig. 10. Evaluation of estimation performance of the CM method when inequation (38) is used as the "cross detector." LPEs and labeling uncertainties are extracted from the clusters "order" 1 and 2, illustrated on top and middle figures. Alg. 2 is the performance evaluator algorithm. The figure at the bottom confirms that the "order detector" in inequation (38) is appropriate. In fact, the associated clustering method removes particle mixing within clusters after objects separation.

OSPA errors. This becomes possible even when extracting cost efficient MMSE [see equation (16)] point estimates thanks to the clustering method, which separates particles in different state of "order." These results support our argument that the design of the proper "cross detector" should produce a decomposition of $p(\boldsymbol{s}_k|\mathbf{Z}_k)$ where each component is unimodal. Also, very accurate estimation of labeling uncertainty is provided, which can be calculated by simply considering the proportion of particles within each order-dependent cluster [see equation (17)].

C. Generalization of the *Cross* Detector to 2–3*D* Objects Settings

The function $K_{2-3D} : \mathbb{R}^{12} \rightarrow \mathbb{R}$ is the counterpart of K_{2-1D} for the 2–3*D* objects case:

$$K_{2-3D} = 2(norm(\boldsymbol{s}'^r_{k-1}) - norm(\boldsymbol{s}'^b_{k-1}))(norm(\boldsymbol{s}'^b_k) - norm(\boldsymbol{s}'^r_k)), \quad (39)$$

where now $\boldsymbol{s}'^p_k = [x^{p,b}_k\ y^{p,b}_k\ z^{p,b}_k\ x^{p,r}_k\ y^{p,r}_k\ z^{p,b}_k]^T$. As $\boldsymbol{s}'^p_k$ and $\Pi(\boldsymbol{s}'^p_k)$ are in different states of "order," K_{2-3D} is also and odd function (in the "order" of $\boldsymbol{s}'^p_k$) which complies with the conditions in equations (33) and (35). The evaluations of the current state and its permuted version are equal in absolute value but different in sign. Therefore, the exact same definition for the "cross detector" from K_{2-1D} and K_{2-2D} cases can be used in the K_{2-3D} case:

$$K_{2-3D} > 0. \quad (40)$$

1) Simulations Results: The results provided by the CM method for the 2–3*D* objects case using inequation (40) as the "cross detector" are shown in Figs. 11 and 12.

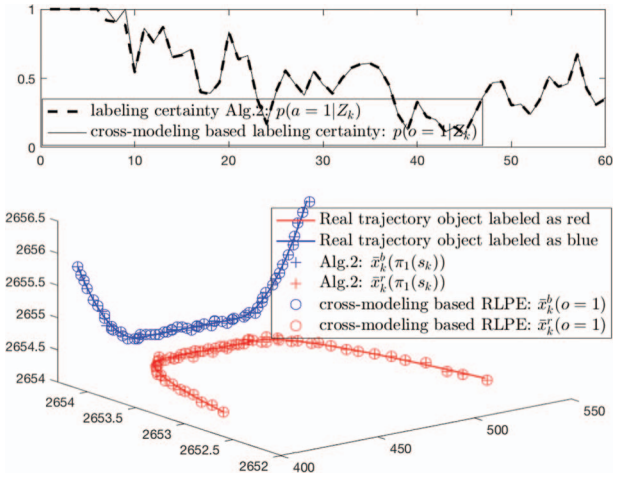


Fig. 11. Evaluation of estimation performance of the CM method when inequation (40) is used as the "order switch" detector. Labeled point estimates are extracted from the cluster "order" 1. Alg. 2 is the performance evaluator algorithm.

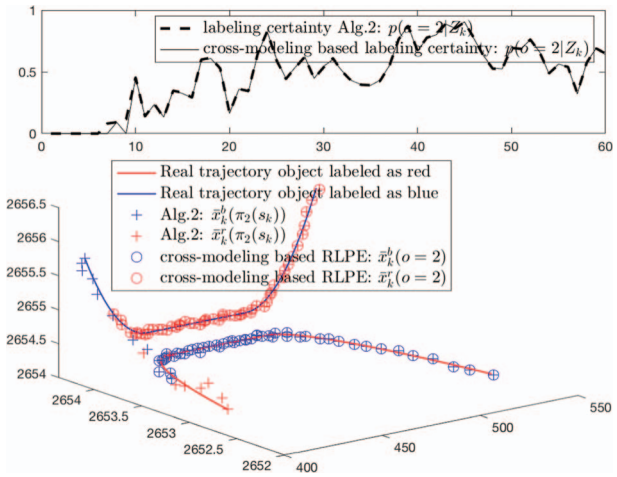


Fig. 12. Evaluation of estimation performance of the CM method when inequation (40) is used as the "cross detector." Labeled point estimates are extracted from the cluster "order" 2. Alg. 2 is the performance evaluator algorithm.

D. Discussion of the Results

The results reveal remarkably accurate estimation performance of the CMT both in the computation of

LPEs and labeling probabilities. This validates the CMT as a convenient solution to answer the questions in Section II also for 3D objects.

VII. CONCLUSIONS

Cross Modeling Tracking has been presented as a DA-free solution to the MOT problem. DA-free solutions are specially relevant in the TBD context, where advantages concerning tracking of low SNR and closely-spaced objects have been reported in previous literature. The main contribution of the paper is the non-trivial generalization of the idea in [17, Section IV] from the 2–1D objects case to the t–MD objects cases.

The underlying problem to solve the MOT problem in TBD has been formulated as: how to decompose a DA-free Bayes posterior so that labeling uncertainty can be characterized. The reviewed method in [17, Section IV] provides an answer based on modeling crosses between objects, but it is only suitable in the 2–1D objects case. A generic formulation of the desired decomposition for the t–MD case is provided. Furthermore, the generalized definition of cross-between-objects has been derived from a meaningful interpretation of the problem in the low dimensional setting.

The paper also revisits the method in [18] and uses it to generate optimal references in order to evaluate estimation performance of the CMT. Simulation results involving challenging closely-spaced objects scenarios have been provided. The results illustrate that the CMT is usable in the general t–MD objects case under the assumption that t is known and constant. Therefore, the proposed solution extends the state-of-the-art of TBD MOT. For the first time in literature, characterization of labeling uncertainty with validated estimation performance and efficient scalability has been provided for seamless use in TBD context.

As for future work, the formulation of the CMT within the RFS framework will be investigated in order to add cardinality estimation capabilities. This is required for the CMT to be declared as a solution for the complete MOT problem, where the assumption of known and constant number of objects does not hold. Additionally, studying the use of the CMT in the context of DBT may lead to interesting advantages. This may be specially the case when comparing with DBT methods in scenarios where DA needs to account for large number of non-negligible hypotheses.

APPENDIX A RELATION BETWEEN THE DATA ASSOCIATION PROBLEM AND THE LABELING PROBLEM

Definitions:

- DA defines hypotheses matching detections and labels.
- Labeling association defines hypotheses matching point estimates and labels.

Applicability:

- DA applies before update.
- Labeling association applies after update and point estimate extraction.

The interesting problem from the application point of view is the labeling problem. The purpose of this appendix is illustrating that:

- DBT trackers only tackle the DA problem. However, once the DA problem is solved, the solution of the labeling problem follows right away: one only needs to perform association dependent updates and extract (labeled) point estimates.

Let us take the following considerations in order to prove that once DA is solved, the solution to the labeling problem follows right away:

- In DBT context, let us consider:
  - Perfect detectability, no false detections.
  - No merged measurements.
  - The measurements do not provide any info about label.

Let us define $\boldsymbol{\varkappa}_k$ as the vector of detections at time step k with explicit indication of correct DA: $\boldsymbol{\varkappa}_k = [\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_t]^T$. The vector $\boldsymbol{\omega} = [\pi_1, \pi_2, ..., \pi_{t!}]^T$ contains the functions to perform all possible permutations of t elements. The tracker cannot access $\boldsymbol{\varkappa}_k$ but an unlabeled version of it, which we denote as $\boldsymbol{z}_k$.

The DBT measurement $\boldsymbol{z}_k$ is a set since different ways of ordering detections define the same measurement. Nevertheless, one can formulate $\boldsymbol{z}_k$ as a vector, $\boldsymbol{z}_k = [\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_t]^T$. Note that the subscripts in $\boldsymbol{\varkappa}_k$ are labels, the subscripts in $\boldsymbol{z}_k$ only define the particular order in which detections are collocated, as this order is random:

$$P(\boldsymbol{z}_k = \pi_m(\boldsymbol{\varkappa}_k)) = P(\boldsymbol{z}_k = \pi_n(\boldsymbol{\varkappa}_k)) \quad \forall\{m, n\} : \{\pi_m, \pi_n\} \in \boldsymbol{\omega}. \tag{41}$$

Let us consider the state vector $\boldsymbol{s}_k$, where the individual states of t objects are staked with explicit indication of labels: $\boldsymbol{s}_k = [\boldsymbol{s}_k^1, \boldsymbol{s}_k^2, ..., \boldsymbol{s}_k^t]^T$. Solving the DA problem requires: generation of DA hypotheses and evaluation of the DA hypotheses. Let us use $\boldsymbol{h}_k$ to note the set of generated DA hypotheses: $\boldsymbol{h}_k = \{(\boldsymbol{a_1}, \boldsymbol{a_2}, ..., \boldsymbol{a_{t!}})\}$, where $\boldsymbol{a_m} = \pi_m(\boldsymbol{z}_k)$, for m =1, 2, .., t!. Note that the subscripts in the vector $\boldsymbol{a_m}$ are not labels, the labels are implicit in the order of $\boldsymbol{a_m}$. For illustrative purposes, consider there are two objects and $\boldsymbol{a_1} = [\boldsymbol{d^2}, \boldsymbol{d^1}]^T$: this means that $\boldsymbol{a_1}$ hypothesizes that the detection which came in second position in $\boldsymbol{z}_k$ was produced by target labeled as 1 and the one which came in first position was produced by the target labeled as 2.

Given the model for the likelihood of the measurements conditioned on the state $l(\boldsymbol{z}_k|\boldsymbol{s}_k)$, a DA dependent

decomposition of $p(\boldsymbol{s}_k|\boldsymbol{Z}_k)$ can be formulated as:

$$p(\boldsymbol{s}_k|\boldsymbol{Z}_k) \propto \sum_{m=1}^{t!} l(\pi_m(\boldsymbol{z}_k)|\boldsymbol{s}_k)p(\boldsymbol{s}_k|\boldsymbol{Z}_{k-1}). \tag{42}$$

From equation (42), one can conclude that in DBT an analytical decomposition relevant for labeling characterization follows from solving the DA problem. In fact, each component in the sum directly relates one DA hypotheses $\pi_m(\boldsymbol{z}_k)$ with its association-dependent posterior $l(\pi_m(\boldsymbol{z}_k)|\boldsymbol{s}_k)p(\boldsymbol{s}_k|\boldsymbol{Z}_{k-1})$. Furthermore, extraction of sufficient statistics from each component in the sum, by minimizing the MMSE, provides a labeled point estimate. Note that a labeled point estimate explicitly hypothesizes one particular association between point estimates and labels. This trivial derivation demonstrates that once DA is solved, the solution of the labeling problem follows right away.

Remarks:

- The analytical evaluation of labeling association hypotheses has a one-to-one relation to the evaluation of DA hypotheses. In particular, the certainty associated to the labeled point estimate extracted from $l(\pi_m(\boldsymbol{z}_k)|\boldsymbol{s}_k)p(\boldsymbol{s}_k|\boldsymbol{Z}_{k-1})$ is:

$$p(\pi_m(\boldsymbol{z}_k)|\boldsymbol{Z}_k) = \frac{\int l(\pi_m(\boldsymbol{z}_k)|\boldsymbol{s}_k)p(\boldsymbol{s}_k|\boldsymbol{Z}_{k-1})d\boldsymbol{s}_k}{\sum_{m=1}^{t!}\int l(\pi_m(\boldsymbol{z}_k)|\boldsymbol{s}_k)p(\boldsymbol{s}_k|\boldsymbol{Z}_{k-1})d\boldsymbol{s}_k}. \tag{43}$$

- Under the assumption that $p(\boldsymbol{s}_k|\boldsymbol{Z}_{k-1})$ can be formulated as one single density $\forall k$, equation (42) does not run into combinatorial explosion over time.
- This analytical derivation is the base of the optimal reference that will be used to evaluate the CMT proposed as the main contribution of this paper.
- Note that the evaluations of $l(\pi_m(\boldsymbol{z}_k)|\boldsymbol{s}_k)$ and $l(\boldsymbol{z}_k|\pi_m(\boldsymbol{s}_k))$ are equivalent. Therefore, equation (42) can be rewritten as:

$$p(\boldsymbol{s}_k|\boldsymbol{Z}_k) \propto \sum_{m=1}^{t!} l(\boldsymbol{z}_k|\pi_m(\boldsymbol{s}_k))p(\boldsymbol{s}_k|\boldsymbol{Z}_{k-1}). \tag{44}$$

APPENDIX B NAIVE GENERALIZATION OF o_k

A naive extension of the definition of "cross" from 2–1D to 2–2D settings will be presented in this appendix. This is not only to illustrate the complexity of the problem but also to familiarize the reader with some consistency checks that the correct extension of "cross" should comply with.

The naive attempt presented in this appendix considers that inequation (32) is the "cross detector" for the 2–2D objects case. After all, a point can be evaluated by the $l2-norm$ function disregarding the dimensionality of the point. Only two formal modifications need to be accounted for when extending inequation (32) from 2–1D to 2–2D settings. First, the $l2-norm$ calculation for the 2–2D objects case becomes:

$$norm(\boldsymbol{s}'_k - \boldsymbol{s}'_{k-1}) = \sqrt{(x^r_k - x^r_{k-1})^2 + (x^b_k - x^b_{k-1})^2 + (y^r_k - y^r_{k-1})^2 + (y^b_k - y^b_{k-1})^2}, \tag{45}$$

where now $\boldsymbol{s}'^{p}_k = [x^{p,b}_k\ y^{p,b}_k\ x^{p,r}_k\ y^{p,r}_k]^T$. Second, the permutation matrix Π for the 2–2D objects case becomes:

$$\Pi = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \tag{46}$$

Given these modifications, the results provided by the CMT for the 2–2D objects case are shown in Fig. 13.

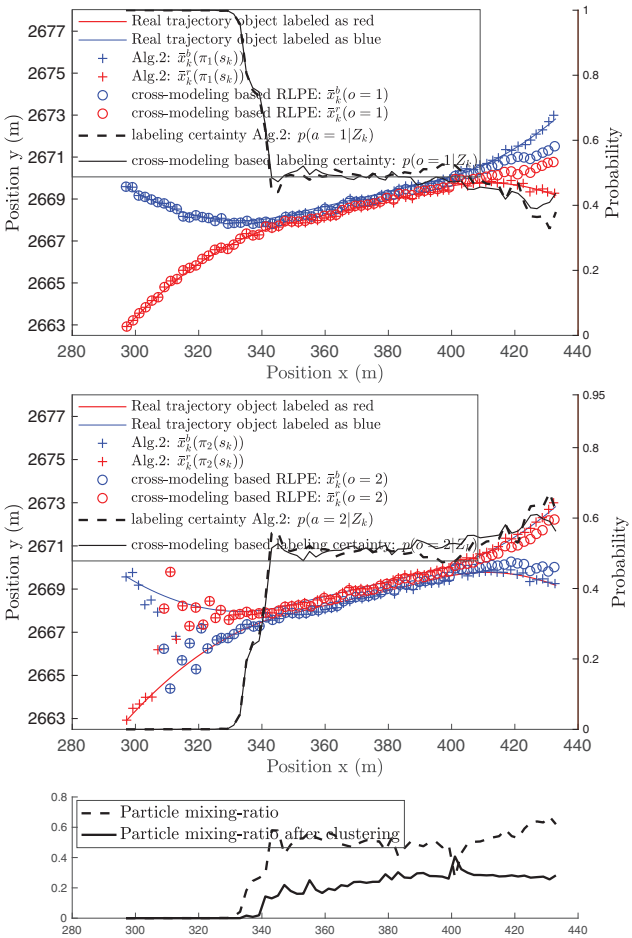


Fig. 13. Evaluation of estimation performance of the CM method using a naive 2–2D "cross" detector. Alg. 2 generates the optimal references.

The results reveal degraded estimation performance of the CMT both in the computation of LPEs and labeling probabilities. In fact, LPEs result in high OSPA errors (specially after objects separation) due to the well known "track coalescence" effect. This undesired effect

is expected if MMSE point estimates are extracted from multimodal densities when labeling uncertainty characterization is not appropriately accounted for.

One can conclude that this attempt to generalize the definition of "cross" is naive. In fact, the use of inequation (32) as the "cross detector" does not remove particle mixing inside each cluster after objects separation. This is illustrated at the bottom part of Fig. 13, where the particle mixing-ratio using CMT clustering, calculated as the ratio between the number of "mixed particles" and the total number of particles, does not drop to zero after target separation.

REFERENCES

[1] E. H. Aoki, P. K. Mandal, L. Svensson, Y. Boers, and A. Bagchi "Labeling uncertainty in multitarget tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 3, pp. 1006–1020, Jun. 2016.

[2] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. Canada: Wiley, 2001.

[3] S. Blackman and R. Popoli *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.

[4] H. Blom and E. Bloem "Permutation invariance in Bayesian estimation of two targets that maneuver in and out formation flight," in *Proc. 12th Int. Conf. Inf. Fusion*, Seattle, 2009, pp. 1296–1303

[5] H. Blom, E. Bloem, Y. Boers, and H. Driessen "Tracking closely spaced targets: Bayes outperformed by an approximation?," in *Proc. 11th Int. Conf. Inf. Fusion*, Cologne, 2008, pp. 1–8.

[6] Y. Boers and J. N. Driessen "Multitarget particle filter track before detect application," *IEE Proc. Radar Sonar Navig.*, vol. 151, no. 6, pp. 351–357, Dec. 2004.

[7] Y. Boers and H. Driessen "The mixed labeling problem in multi target particle filtering," in *Proc. 10th Int. Conf. Inf. Fusion*, Quebec, 2007, pp. 1–7.

[8] Y. Boers, E. Sviestins, and H. Driessen "Mixed labelling in multitarget particle filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 2, pp. 792–802, Apr. 2010.

[9] D. F. Crouse, P. Willett, and Y. Bar-Shalom "Generalizations of Blom and Bloem's PDF decomposition for permutation-invariant estimation," *Proc. 2011 IEEE Int. Conf. Acoust. Speech Signal Process.*, 2011, pp. 3840–3843

[10] D. F. Crouse, P. Willett, and Y. Bar-Shalom "Developing a real-time track display that operators do not hate," in *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3441–3447, Jul. 2011.

[11] D. F. Crouse, P. Willett, L. Svensson, D. Svensson, and M. Guerriero "The Set MHT," in *14th Int. Conf. Inf. Fusion*, Chicago, IL, 2011, pp. 1–8.

[12] M. Ekman, E. Sviestins, L. Sjoberg, Y. Boers, and H. Driessen "Particle filters for tracking closely spaced targets," in *Proc. 10th Int. Conf. Inf. Fusion*, Quebec, 2007, pp. 1–8.

[13] T. Fortmann, Y. Bar-Shalom, and M. Scheffe "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE J. Ocean. Eng.*, vol. 8, no. 3, pp. 173–184, Jul. 1983.

[14] A. F. García-Fernández "Track-before-detect labeled multi-bernoulli particle filter with label switching," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 5, pp. 2123–2138, Oct. 2016.

[15] A. F. García-Fernández, M. Morelande, and J. Grajal "Particle filter for extracting target label information when targets move in close proximity," in *Proc. 14th Int. Conf. Inf. Fusion*, Chicago, 2011, pp. 1–8.

[16] A. F. García-Fernández, L. Svensson, and M. R. Morelande "Multiple target tracking based on sets of trajectories," *IEEE Trans. Aerosp. Electron. Syst.* vol. 56, no. 3, pp. 1685–1707, Jun. 2020.

[17] C. M. Leon, H. Driessen, and P. K. Mandal "Efficient characterization of labeling uncertainty in closely-spaced targets tracking," in *Proc. 19th Int. Conf. Inf. Fusion*, Heidelberg, 2016, pp. 449–456.

[18] C. M. Leon and H. Driessen "Evaluation of labeling uncertainty in multiple target tracking with track-before-detect radars," in *Proc. 22nd Int. Conf. Inf. Fusion*, Ottawa, 2019, pp. 1–8.

[19] R. Mahler "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.

[20] M. R. Morelande and A. Zhang "Uniform sampling for multiple target tracking," in *Proc. 14th Int. Conf. Inf. Fusion*, Chicago, 2011, pp. 1–7.

[21] D. Reid "An algorithm for tracking multiple targets," *IEEE Trans. Automat. Contr.*, vol. 24, no. 6, pp. 843–854, 1979.

[22] B. Ristic, S. Arulampalam, and N. Gordon *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Norwood, MA, USA: Artech House, 2004.

[23] L. Svensson and M. Morelande "Target tracking based on estimation of sets of trajectories," in *Proc. 17th Int. Conf. Inf. Fusion*, Salamanca, 2014, pp. 1–8.

[24] Y. Xia, L. Svensson, A. F. García-Fernández, K. Granström, and J. L. Williams "Backward simulation for sets of trajectories," in *Proc. 23rd Int. Conf. Inf. Fusion*, virtual, 2020, pp. 1–8.

Carlos Moreno Leon received his MSc in Telecommunications Engineering in 2013 from the University of Alcalá (UAH) in Madrid. From 2012 to 2014 he was employed with the Polytechnic University of Madrid (UPM) as a research assistant in the Signal, Systems and Radiocommunications department. From 2014 to 2017 he held a European Commission Marie Curie research fellowship while working at Thales Nederland BV. Since 2017 he is with the Cognitive Radar department in the German Fraunhofer Institute for High Frequency Physics and Radar Techniques (FHR). Parallel to his work at Fraunhofer FHR, he is pursuing a PhD at TU-Delft since November 2017 (in the Microwave Signals Sensor and Systems group). He has been involved in several international research and development projects in the area of signal/data processing with applications in Wireless Sensor Networks and Radar Technology both within civilian and military contexts.

Hans Driessen obtained the MSc and PhD degree in 1987 and 1992, respectively, both from the department of Electrical Engineering at the TU-Delft. Since then he has been employed with Thales Nederland BV. He has been and is still involved in various (international) research projects and radar development programs in the area of signal/data processing and radar management. Since January 2015, he additionally holds a part-time position as associate professor at the EEMCS faculty in the Microwave Signals Sensor and Systems group in the field of radar systems; waveforms and processing. His interest is in the practical application of detection, estimation, information and control theory to various problems in sensor systems.

Alexander Yarovoy (FIEEE'2015) graduated from the Kharkov State University, Ukraine, in 1984 with the Diploma with honor in radiophysics and electronics. He received the Candidate Phys. & Math. Sci. and Doctor Phys. & Math. Sci. degrees in radiophysics in 1987 and 1994, respectively. In 1987 he joined the Department of Radiophysics at the Kharkov State University as a Researcher and became a Full Professor there in 1997. From September 1994 through 1996 he was with Technical University of Ilmenau, Germany as a Visiting Researcher. Since 1999 he is with the Delft University of Technology, the Netherlands. Since 2009 he leads there a chair of Microwave Sensing, Systems and Signals. His main research interests are in high-resolution radar, microwave imaging and applied electromagnetics (in particular, UWB antennas). He has authored and co-authored more than 450 scientific or technical papers, six patents and fourteen book chapters. He is the recipient of the European Microwave Week Radar Award for the paper that best advances the state-of-the-art in radar technology in 2001 (together with L.P. Ligthart and P. van Genderen) and in 2012 (together with T. Savelyev). In 2010 together with D. Caratelli Prof. Yarovoy got the best paper award of the Applied Computational Electromagnetic Society (ACES). Prof. Yarovoy served as the General TPC chair of the 2020 European Microwave Week (EuMW'20), as the Chair and TPC chair of the 5th European Radar Conference (EuRAD'08), as well as the Secretary of the 1st European Radar Conference (EuRAD'04). He served also as the co-chair and TPC chair of the Xth International Conference on GPR (GPR2004). He served as an Associated Editor of the *International Journal of Microwave and Wireless Technologies* from 2011 till 2018 and as a Guest Editor of five special issues of the IEEE Transactions and other journals. In the period 2008-2017 Prof. Yarovoy served as Director of the European Microwave Association (EuMA).

ML-PMH Tracking in Three Dimensions Using Cluttered Measurements From Multiple Two-Dimensional Sensors

ZACHARIAH SUTTON
PETER WILLETT
TIM FAIR
YAAKOV BAR-SHALOM

The maximum-likelihood probabilistic multi-hypothesis tracker (ML-PMHT) is a tracking method whose flexibility and scalability derive from relinquishing the assumption that each target emits at most one "hit" per scan of the sensor. This is an ML method that essentially reduces to an optimization problem—recursively maximizing a likelihood function that is simple to evaluate given a batch of observations. Unlike maximum *a posteriori* or minimum mean squared error (MMSE) trackers, this likelihood maximization tracker requires neither prior knowledge about target motion nor measurement association, making it conceptually easy to work with. Here, this method is used to track targets in a three-dimensional "global" space with observations provided by multiple two-dimensional sensors placed throughout the global space. Since the observation model is non-linear, the likelihood maximization is done via hill climbing. For this purpose, we also address the issue of "hill finding." Due to the presence of clutter in the measurement model, the likelihood is a multi-modal function of the parameter space. That is, there are multiple hills in the likelihood function, and it is of great advantage to the tracker to initialize the hill climber close to the right hill—the one whose peak is the global maximum. In this work, we present a data-driven method of initializing the hill climber based on the received observations.

Manuscript received August 19, 2021; revised September 27, 2021; released for publication December 1, 2021.

Z. Sutton, P. Willett, and Y. Bar-Shalom are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA E-mail: zachariah.sutton@uconn.edu, peter.willett@uconn.edu, yaakov.bar-shalom@uconn.edu
T. Fair is from Toyon Research Corporation, E-mail: tfair@toyon.com

This work was partially supported by AFOSR under contract FA9500-18-1-0463.

I. BACKGROUND

The maximum-likelihood probabilistic multihypothesis tracker (ML-PMHT) is an ML target tracking paradigm that is convenient in cases where data association—the measurement-to-target assignment processes prior to updating the estimate—involves significant numerical complexity, generally (but not always) due to heavy clutter. In some settings, it is possible that a single target will result in multiple measurements at a particular sensor and time (tracking of "extended objects," for instance). In such cases, filters that employ "hard" data association (the JPDA [20] and random finite set filters such as the multi-Bernoulli [6], [15]) will be sub-optimal since they make the fundamental assumption that each target being tracked produces at most one measurement per sensor per time step. In contrast, the likelihood function used in the ML-PMHT is formulated by considering each measurement individually, and applying a probability mass function over the possible measurement generating processes (targets and clutter). That is, instead of assuming that a particular measurement has come from a particular target and evaluating the measurement likelihood with that assignment, the ML-PMHT formulates the measurement likelihood with a "soft" assignment that accounts for uncertainty as to the process from which a particular measurement originated. This formulation naturally allows for the possibility that a target has originated multiple measurements in a single "scan" of a sensor. Along with being a better representation of reality in some settings, the soft assignment also avoids the computational bookkeeping cost of the hard assignment problem that the data association filters must solve for each scan with relatively expensive routines like Murty's k-Best Assignment Algorithm [12]. Thus, the ML-PMHT approach may also be desirable in some settings where computational cost is a consideration.

The ML-PMHT likelihood formulation is borrowed from the PMHT framework [7], [10]. The ML-PMHT differs, however, in that it treats the target state (joint target state in the case of multiple targets) as an unknown deterministic parameter, and obtains an ML estimate of the parameter based on batches of measurement scans. It has shown especially good performance in scenarios with high levels of clutter [14], [21].

This work will use the ML-PMHT to perform data-batch-based tracking of targets in a three-dimensional "global" space based on multiple passive sensors that return two-dimensional measurements. A generalized measurement model is presented that can be adapted to any type of sensor that returns measurements that can be transformed into lines-of-sight. Some common sensor types that could be used with this model are focal plane arrays (cameras) with measurements given in the two-dimensional image space, or passive radars that return azimuth and elevation angles (or azimuth and elevation angle sines).

It is assumed that all sensors report to a central processor that performs likelihood maximization based on all observed data. The central optimization reveals another benefit of the ML-PMHT in that data from multiple sensors are naturally included in the likelihood formulation in a simple linear sum manner. While sequential updating over sensors is a common practice in data association filters, it is theoretically sub-optimal [20]; and optimality with multiple sensors according to the rules of "hard" data association is computationally costly.

The ML-PMHT reduces to a conceptually straightforward optimization problem where likelihood maximization happens over a multi-dimensional parameter space. However, an important practical consideration in optimization problems is initialization: how to pick the "initial guess" for the parameter value. Initialization is particularly important in the tracking setting since we assume the presence of clutter measurements, which makes the likelihood multi-modal. That is, coincidental "patterns" in clutter measurements can lead to false maximums in the likelihood value that gradient-based maximizers will reach in error if not initialized carefully. Although the global maximum tends to occur at the true parameter value, a maximizer must be initialized sufficiently close to the global maximum in order to reach it. For this purpose, a "hill finding" method is presented where received measurements are used to identify statistically significant points in the global parameter space that can be used to initialize the maximization. This hill finding routine is conceptually separate from the ML-PMHT, and is perhaps the most novel contribution of this work. The method represents a means of identifying pairs of line-of-sight measurements from separate sensors that strongly correlate to a single point in three-dimensional space. Thus, it could theoretically be used in other settings where one would wish to identify points in three-dimensional space that are statistically supported by lower-dimensional measurements. (It could inform the "target birth" process in the multi-Bernoulli filter, for example.)

The benefit of central data processing and the resulting ability to perform the "hill finding" is demonstrated by comparing the method to a decentralized option where ML estimates are obtained individually by each sensor, then fused in the global space. It is shown that centralized optimization has a significant advantage in settings with low target visibility. A comparison is also made to the joint probabilistic data association filter (JPDAF).

The paper is structured as follows. The models used in the work are presented in Section II, including the target parameter model, the measurement model, and the model of the geometric arrangement of sensors. The ML-PMHT likelihood formula is given in Section III. The "hill finder" is presented in Section IV, and a step-by-step summary of the overall ML-PMHT method is given. Simulated results are shown in Section V.

II. MODELING ASSUMPTIONS

The model assumes a three-dimensional global space in which targets are to be tracked. Measurements are received from a group of N^{s} sensors distributed around the global space. This work uses a conventional Cartesian coordinate system in the global space, but in theory it could be replaced by a local north-east-down reference frame or any other space where the following conditions are fulfilled:

1) The motion of targets can be (approximately) parameterized in the space.
2) The *pose* of every sensor is known in the space.

The pose of a sensor parameterizes the transformation between the global coordinate system and the sensor's coordinate system. The sensors are assumed to have six degrees of freedom (DOFs)—three translational (location) and three rotational (pointing). So the pose consists of six known parameters for each sensor. If a sensor's pose changes over time, it is assumed to be known for each point in time that a measurement is received. The conventions used for the pose and the resulting transformations are discussed in more detail in Section II.C.

A. Target Motion

The user must choose a batch size parameter N^{b}, which is the number of scans from each sensor that will be used in the likelihood evaluation. It is assumed that, for all targets, the true target motion can be reasonably approximated by a constant velocity model over the duration of the batch. That is, for any discrete global time index k, the motion of target j over the past $N^{\mathrm{b}}-1$ sampling periods (N^{b} sampling points) is given by

$$\mathbf{X}_{j,\ell} \simeq (\mathbf{X}_{j,k} - \mathbf{X}_{j,k-N^{\mathrm{b}}+1})\frac{n}{N^{\mathrm{b}}-1} + \mathbf{X}_{j,k-N^{\mathrm{b}}+1} \qquad (1)$$

$$n = 0, \ldots, N^{\mathrm{b}}-1,$$

where $\mathbf{X}_{j,\ell}$ denotes a three-element column vector containing the target's position in the global Cartesian space at global time step ℓ and n is a local time index such that $\ell = k - N^{\mathrm{b}} + 1 + n$.

A constant sampling period is assumed here, but there is no loss of generality. With the batch size chosen, the motion in (1) is entirely parameterized by $\mathbf{X}_{j,k-N^{\mathrm{b}}+1}$ and $\mathbf{X}_{j,k}$—the positions of target j at the start and end times of the batch. Thus, the motion to be estimated via likelihood maximization can be described with six parameters for each target. For a scenario with N^{t} targets

present, form a parameter vector

$$\boldsymbol{\mathcal{X}}^k \triangleq \begin{bmatrix} \mathbf{X}_{1,k-N^{\mathrm{b}}+1} \\ \mathbf{X}_{1,k} \\ \vdots \\ \mathbf{X}_{N^{\mathrm{t}},k-N^{\mathrm{b}}+1} \\ \mathbf{X}_{N^{\mathrm{t}},k} \end{bmatrix}. \tag{2}$$

We ultimately will be maximizing the log likelihood function over this vector given $\boldsymbol{\mathcal{Z}}^k$—the batch of measurements up to and including time step k—to obtain the estimate

$$\boldsymbol{\mathcal{X}}^k_{\mathrm{ML}} = \arg\max_{\boldsymbol{\mathcal{X}}^k \in \mathbb{R}^{6N^{\mathrm{t}}}} \mathcal{L}(\boldsymbol{\mathcal{X}}^k; \boldsymbol{\mathcal{Z}}^k). \tag{3}$$

The definition of $\boldsymbol{\mathcal{Z}}^k$ and the formulation of the likelihood function $\mathcal{L}(\cdot)$ are discussed later. The maximization is mentioned here to emphasize that the dimensionality of the space over which maximization is performed increases by 6 for each additional target. This is a computational consideration in a practical application.

In theory, this likelihood maximization requires only the constant velocity assumption over the course of any particular batch. However, it will be of use if the user has some more prior knowledge about target motion. Specifically, if there is knowledge available about the range of possible target speeds, it will prove useful in the initialization of the hill climber as discussed in Section IV.A.

Note that the particular six-parameter motion model used here is not the only model compatible with batch tracking. One could also choose—at the cost of computation time—to use a nine-parameter (initial position, initial velocity, and acceleration) model or any other method of parameterizing the target motion over the duration of the batch.

B. Measurement Model

The tracker developed in this work uses line-of-sight measurements. That is, it is assumed that each and every sensor returns some form of two-element measurements that can be used to parameterize a line-of-sight beginning at the origin of its own coordinate system and extending infinitely in the direction of sight. The line-of-sight measurement model is a fundamental feature of this work. However, there are multiple types of two-dimensional measurements that provide a line-of-sight, which allows the sensor type to remain ambiguous. Simple passive radar models directly provide line-of-sight measurements. Also, a point in the image space of a camera can be converted into a line-of-sight given the camera model. The simplicity of radars in this context renders them rather uninteresting. Thus, cameras are assumed in the peripheral theoretical modeling and verification in this work, without loss of generalization in the fundamental aspects of the work (likelihood maximization). This section discusses the relationship between image space and line-of-sight measurements within the coordinate system of a sensor's reference frame. The transformation between a sensor's coordinate system and the global coordinate system is discussed in Section II.C.

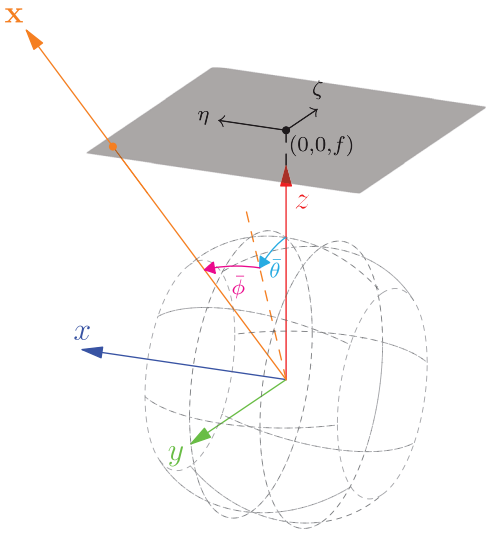


Fig. 1. Illustration of the sensor reference frame convention used in this work. Two different measurement types are shown. Given the coordinates of the vector $\mathbf{x}$, one could solve for $\bar{\theta}$ and $\bar{\phi}$, which is the azimuth–elevation measurement model assumed throughout this work. Given a point in the image plane (shaded) at a known distance f from the origin (where f has the same length units as η, ζ coordinate system), one could also calculate the corresponding azimuth and elevation angles.

When camera images are used with this algorithm, it is necessary to first perform measurement extraction. The assumption when using cameras as sensors is that targets have a contrasting appearance to the background. The extraction must find significantly bright or dark spots in an image, and condense each spot down to a point in the image space of the camera, which, in turn, can be converted into line-of-sight measurements via the camera model. Methods of extracting these measurements from images are discussed in [3] and [11]. The signal-to-noise ratio (SNR) of the scenario is an important factor in the measurement extraction step. Note that a significant portion of [3] is dedicated to defining the SNR. The extraction process will have a certain probability of extracting "false" measurements, which we will refer to as clutter.

Once measurement extraction has been performed on the images, the resulting measurements must be converted from the image space into lines-of-sight. Refer to Fig. 1 for a summary of this conversion. The camera model used here is the pinhole projection model [5], [23]. The image is treated as a plane parallel to the x–y plane of the sensor coordinate frame, set at some non-zero focal distance f along the sensor z axis. Let the coordinate system in the image plane be denoted by (η, ζ), which is centered on the sensor z axis, and has directional convention that agrees with the typical row–column format of images. Then a point $[\eta, \zeta]'$ in the image plane coordinate system has location $[\eta, -\zeta, f]'$ in the coordinate

system of the sensor's reference frame. Note that the image plane is treated as a continuous space (as opposed to a quantized space) since the measurement extraction process yields continuous values. Azimuth–elevation angles are used for parameterizing line-of-sight measurements. The azimuth angle θ is taken in the sensor y–z plane, with zero defined as the positive z axis and the positive direction defined as from the positive z axis to the positive y axis. The elevation angle ϕ is the angle between the sensor y–z plane and the positive x axis, defined as zero on the y–z plane and positive toward the positive x axis. Under this choice of convention, a given point in the image plane $[\eta, \zeta]'$ yields line-of-sight angles

$$\theta = \tan^{-1}\left(\frac{-\zeta}{f}\right), \tag{4}$$

$$\phi = \tan^{-1}\left(\frac{\eta}{\sqrt{\zeta^2 + f^2}}\right). \tag{5}$$

Note that this particular camera model is somewhat simplistic. It does not account for image distortion or other practical effects. If the user has a more accurate model of the cameras, it will be compatible with this algorithm so long as it provides a way to obtain lines-of-sight from points in the image. Since the algorithm ultimately works with line-of-sight measurements, this writing will occasionally use the general term "measurements" when referring to azimuth–elevation measurements.

Overall, it is assumed that at a particular time, each sensor i returns a set of line-of-sight measurements—potentially after conversion with (4) and (5)—which includes any target-originated measurements along with any measurements originating from the clutter process. That is, the set of measurements returned by sensor i at time ℓ can be denoted

$$\mathbf{Z}_{i,\ell} = \{\mathbf{z}_{i,\ell,m}\}_{m=1}^{N_{i,\ell}^{z}} = \left\{\begin{bmatrix}\theta_{i,\ell,m}\\ \phi_{i,\ell,m}\end{bmatrix}\right\}_{m=1}^{N_{i,\ell}^{z}}, \tag{6}$$

where each $\mathbf{z}_{i,\ell,m}$ is a two-element column vector and $N_{i,\ell}^{z}$ is the number of measurements at the current time/sensor.

It will be helpful later on, during formulation of the likelihood function, to have a simple expected value parameterization for the number of clutter and target-originating measurements. The expected numbers of received measurements are fundamentally tied to the measurement extraction process, which is left non-specific for most of this work. Let us assume some general extraction process such that $\lambda_{i,\ell}$ and $\varphi_{i,\ell}$ are the expected numbers of clutter measurements and target-originated measurements, respectively, in the scan from sensor i at time ℓ. For the sake of more generality, allow each individual target j to originate a potentially unique expected number of measurements $\varphi_{i,j,\ell}$ such that

$$\varphi_{i,\ell} = \sum_{j} \varphi_{i,j,\ell}. \tag{7}$$

For purposes of simulation later in this work, it will be assumed that the number of clutter measurements is Poisson with some expected value λ_i. It will be further assumed that, independently for each target present, sensor i either reports a single measurement with "detection probability" $p_{\mathrm{d},i}$ or "misses" the target. Thus, the expected number of target-originated measurements from each target is $p_{\mathrm{d},i}$, and $\varphi_{i,\ell} = p_{\mathrm{d},i}N^{\mathrm{t}}$ for all ℓ. These assumptions are made to fit with a typical model used in other trackers for the sake of comparison. However, one of the main benefits of the ML-PMHT likelihood formulation is that it is more flexible than trackers that consider one-to-one data associations. Whereas the JPDA and its derivative algorithms must make the fundamental assumption that each target produces at most one measurement per scan, the ML-PMHT formulation requires no such assumption. Some data association tracking methods do exist for extended targets (targets that produce more than one measurement in a single scan), and usually involve recursive estimation of properties (e.g., shape, size) of targets. The ML-PMHT offers a relatively cheap way around this extra estimation for the case when the extended target measurement assumption is true, but the shape/size properties of targets are not of particular interest. For instance, a simple model could assume that the number of measurements originating from target j is Poisson random number with expected value $\varphi_{i,j}$.

It is assumed that any target-originated measurements have a random additive measurement error. Since the ML-PMHT is an objective function optimization problem, the parameterization of the measurement error is somewhat flexible: Any objectively computable error probability density can be used. This work will use the typical Gaussian error assumption. That is, if a particular measurement $\mathbf{z}_{i,\ell,m}$ originates from target j, then it is a random vector given by

$$\mathbf{z}_{i,\ell,m} = \bar{\mathbf{z}}_{i,j,\ell} + \boldsymbol{\nu}_{i,j,\ell}, \tag{8}$$

where $\bar{\mathbf{z}}_{i,j,\ell}$ is the noiseless (zero-error) measurement by sensor i due to target j at time ℓ. Given the three-element vector $\mathbf{x}_{i,j,\ell}$ representing the Cartesian (x–y–z) position of target j *in the sensor i coordinate frame* at time ℓ, the noiseless measurement for the azimuth–elevation model is given by

$$\bar{\mathbf{z}}_{i,j,\ell} = \begin{bmatrix}\bar{\theta}_{i,j,\ell}\\ \bar{\phi}_{i,j,\ell}\end{bmatrix} = \begin{bmatrix}\tan^{-1}\left(\frac{y_{i,j,\ell}}{z_{i,j,\ell}}\right)\\ \tan^{-1}\left(\frac{x_{i,j,\ell}}{\sqrt{y_{i,j,\ell}^2 + z_{i,j,\ell}^2}}\right)\end{bmatrix}, \tag{9}$$

where $x_{i,j,\ell}$, $y_{i,j,\ell}$, $z_{i,j,\ell}$ are the individual components of $\mathbf{x}_{i,j,\ell}$. For a visual representation of this measurement model refer to Fig. 1 and treat the orange vector as the target position. The term $\boldsymbol{\nu}_{i,j,\ell}$ in (8) is a two-element multivariate random Gaussian vector with distribution

$$\boldsymbol{\nu}_{i,j,\ell} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{i,\ell}), \tag{10}$$

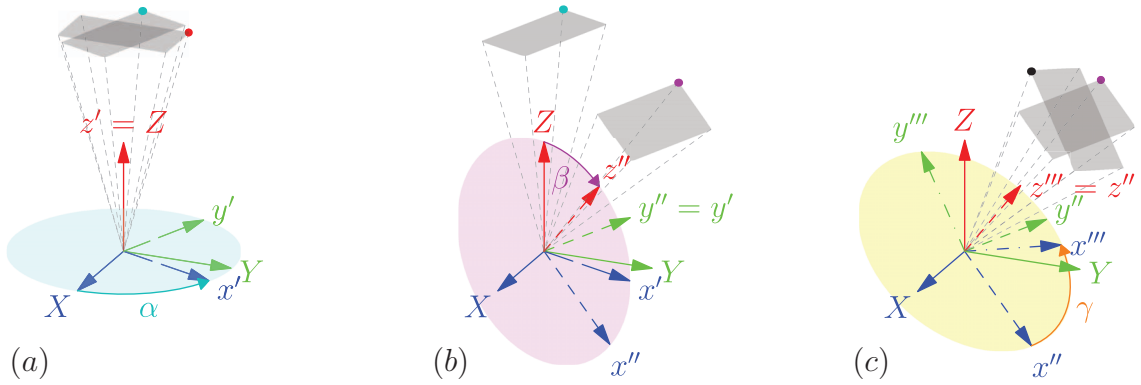


Fig. 2. Step-by-step illustration of a three-DOF proper Euler rotation. The total rotation is the result of applying three elemental rotations successively. The specific convention shown here and assumed in this work is a z–y–z intrinsic rotation. The focal plane shown in the figures matches the orientation of the one shown in Fig. 1. The origin of the (u, v) coordinate system is marked by a dot. The rotations bring the plane from the orientation marked by the red dot in (a) to the orientation marked by the black dot in (c). Orientations that are marked with a common color are the same. Colored circles attempt to show planes of rotation.

where, in general, the covariance matrix $\mathbf{R}_{i,\ell}$ is allowed to change from one sensor to the next and one time step to the next. Measurement errors are assumed to be independent between sensors and between time steps.

Clutter measurements are assumed to be uniformly distributed in the measurement space. It has been assumed in this work that each sensor has a limited field of view, modeled by setting limits on the azimuth and elevation angles symmetrically around zero. Then for a sensor i with a total azimuth field of view W_i^θ and total elevation field of view W_i^ϕ, the spatial distributions of the individual components of clutter measurements are

$$\theta_i^{\text{clutter}} \sim \mathcal{U}(-W_i^\theta/2,\ W_i^\theta/2), \tag{11}$$

$$\phi_i^{\text{clutter}} \sim \mathcal{U}\left(-W_i^\phi/2,\ W_i^\phi/2\right), \tag{12}$$

where, in general, different sensors are allowed to have differently sized fields of view—hence the indexing with i. For sensors with reasonably narrow fields of view (within the range of realistic cameras), the uniform distributions in azimuth and elevation result in image space measurements that are very close to uniform in the image plane. See Figs. 7 and 8 for a visual example of measurements in an image plane resulting from this model.

The total measurement batch $\mathcal{Z}^k$ in (3) can be expressed as

$$\mathcal{Z}^k = \{\{\mathbf{Z}_{i,\ell}\}_{i=1}^{i=N^{\text{s}}}\}_{\ell=k-N^{\text{b}}+1}^{\ell=k}. \tag{13}$$

Or, in words, $\mathcal{Z}^k$ is the set of all subsets of measurements (both target and clutter originated) from all sensors for the most recent N^{b} sample times (up to and including the current estimation time k).

C. Sensor-World Setup

Notice that in Fig. 1 and in the formulas in (8) and (9), it is assumed that $\mathbf{x}_{i,j,\ell}$—the three-dimensional Cartesian position of target j in the sensor i reference frame at time ℓ—is given. Since the likelihood in (3) is being maximized over target motion parameters given in the *global* reference frame, the relationship between the global reference frame and each sensor reference frame must be defined.

Let the axes of the global coordinate system be denoted $(X$–Y–$Z)$ and those of the sensor coordinate system $(x$–y–$z)$. The *pose* of the sensor is the position and orientation of its reference frame relative to the global reference frame, defined such that at a pose of zero, the two coordinate systems are one and the same.

Let the three rotational DOFs be described by the angles $\alpha_{i,\ell}$, $\beta_{i,\ell}$, and $\gamma_{i,\ell}$, where the time index ℓ is noted since the rotation of a sensor can change with time. There are multiple choices of convention for the actual meaning of these angles. To be exact, there are 12 unique ways to describe every possible orientation in terms of three angles.

The rotation convention used here is illustrated in Fig. 2, where, for convenience, a particular sensor at a particular time is considered and the indexing is momentarily dropped. The sensor coordinate system is initially aligned with the global coordinate system. The overall rotation is the combination of three intrinsic rotations performed sequentially.

(a) A rotation by α around the Z axis results in the new coordinate system $(x'$–y'–$z')$.

(b) Then a rotation by β around the y' axis to obtain $(x''$–y''–$z'')$.

(c) Finally, a rotation by γ around the z'' axis gives the fully rotated coordinate system $(x'''$–y'''–$z''')$.

Here, the positive direction for all rotations is given by the "right-hand rule." This describes what is commonly called an z–y–z intrinsic rotation. Here z–y–z refers to the sequence of rotation axes, and intrinsic refers to the fact that successive rotations are performed around the axes of the rotating coordinate system (sensor coordinate system) itself as opposed to rotating

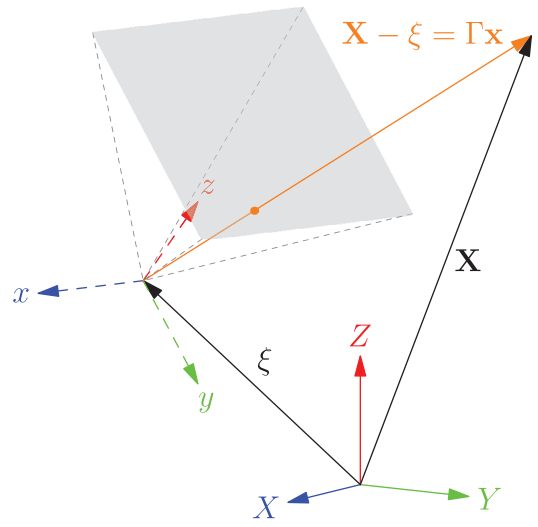


Fig. 3. Illustration of sensor-world setup for a particular sensor, target, and time step. Given target location $\mathbf{X}$ in the global (X–Y–Z) coordinates and sensor pose information $\boldsymbol{\Gamma}$, $\boldsymbol{\xi}$, the zero-error measurements $\bar{\theta}$, $\bar{\phi}$ in (9) are calculated by first evaluating $\mathbf{x}$—the target location in sensor (x–y–z) coordinates.

around the axes of the fixed global coordinate system. There is nothing particularly special about the choice of the z–y–z convention; in light of the camera model in Fig. 1, it is merely a way that one could conceivably go about orienting such a camera in a practical scenario. If the rotation angles are in some other convention, they will work just as well, and only the rotation matrix will change. For a comprehensive description of Euler rotations, see [22].

Referring now to Fig. 3, let the three-dimensional Cartesian position of sensor i at time step ℓ be a vector denoted by $\boldsymbol{\xi}_{i,\ell}$. After rotating the sensor coordinate system, the final sensor coordinate system is given by translating the origin of the fully rotated system into the point $\boldsymbol{\xi}_{i,\ell}$.

Now, given $\mathbf{X}_{j,\ell}$—a position vector for target j at time ℓ in the *global* reference frame—the resulting target position in the reference frame of sensor i is given by the inversing transformation

$$\mathbf{x}_{i,j,\ell} = \boldsymbol{\Gamma}_{i,\ell}^{-1}(\mathbf{X}_{j,\ell} - \boldsymbol{\xi}_{i,\ell}). \tag{14}$$

Here, $\boldsymbol{\Gamma}_{i,\ell}$ is the 3×3 rotation matrix that transforms Cartesian points from the fixed global frame to the rotated (but non-translated) reference frame. For the particular rotation convention used in this work, the rotation matrix is given by

$$\boldsymbol{\Gamma}_{i,\ell} = \begin{bmatrix} \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & -\cos\gamma\sin\alpha - \cos\alpha\cos\beta\sin\gamma & \cos\alpha\sin\beta \\ \cos\alpha\sin\gamma + \cos\beta\cos\gamma\sin\alpha & \cos\alpha\cos\gamma - \cos\beta\sin\alpha\sin\gamma & \sin\alpha\sin\beta \\ -\cos\gamma\sin\beta & \sin\beta\sin\gamma & \cos\beta \end{bmatrix}, \tag{15}$$

where the (i, ℓ) indexing is removed from the angles for convenience. Note that $\boldsymbol{\Gamma}_{i,\ell}$ entirely describes the rotation of sensor i at time ℓ. That is, the poses of all sensors can be recorded as the set of data matrices

$$\mathbf{S} = \{\boldsymbol{\Gamma}_{i,\ell},\ \boldsymbol{\xi}_{i,\ell}\}_{i=1}^{N^s}, \quad \forall \ell. \tag{16}$$

III. LIKELIHOOD EVALUATION

A fundamental feature of the ML-PMHT is that no hard limit is assumed for the maximum number of measurements originating from any one target in any one sensor at any one sample time. This is a significant departure from the assumptions made in data association filters where various one-to-one measurement-to-target assignment events are enumerated and considered. This modeling relaxation allows the ML-PMHT formulation to consider any particular measurement (indexed m) independently from all other measurements, and assign a prior probability mass function over the set of possible measurement generating processes (clutter and all targets)

$$\Pi_{i,\ell} = (\pi_{i,j,\ell})_{j=0}^{N^t}, \quad \text{s.t.} \quad \sum_{j=0}^{N^t} \pi_{i,j,\ell} = 1, \tag{17}$$

where $\pi_{i,j,\ell}$ is the prior probability that any particular measurement in the scan from sensor i originated from process j, and $j = 0$ indicates the clutter process. For now, allow the value of the priors to be ambiguous. Methods for setting the priors are discussed in Section III.A.

The most conveniently scaled statistic to maximize is the log-likelihood ratio (LLR) of the target state $\boldsymbol{\mathcal{X}}^k$ based on the measurement batch $\boldsymbol{\mathcal{Z}}^k$. By definition, the LLR is given by

$$\mathcal{L}(\boldsymbol{\mathcal{X}}^k; \boldsymbol{\mathcal{Z}}^k) = \ln\left(\frac{p(\boldsymbol{\mathcal{Z}}^k|\boldsymbol{\mathcal{X}}^k)}{p(\boldsymbol{\mathcal{Z}}^k|\boldsymbol{\emptyset})}\right), \tag{18}$$

where $p(\boldsymbol{\mathcal{Z}}^k|\boldsymbol{\emptyset})$ represents the probability density function (pdf) of the measurement batch given that no targets are present—the pdf of the entire batch given that everything is clutter-generated. Under the measurement independence assumptions and the product to sum logarithm property, (18) can be written as

$$\mathcal{L}(\boldsymbol{\mathcal{X}}^k; \boldsymbol{\mathcal{Z}}^k) = \sum_{i=1}^{N^s} \sum_{\ell=k-T^b}^{k} \sum_{m=1}^{N_{i,\ell}^z} \ln\left(\frac{p(\mathbf{z}_{i,\ell,m}|\boldsymbol{\mathcal{X}}^k)}{p(\mathbf{z}_{i,\ell,m}|\boldsymbol{\emptyset})}\right), \tag{19}$$

where $N_{i,\ell}^z$ is the number of measurements in the scan of sensor i at time step ℓ. The term $p(\mathbf{z}_{i,\ell,m}|\boldsymbol{\emptyset})$ is the pdf of a single measurement given that it originated from the clutter process. Under the simplifying assumption in (11) and (12), this pdf is uniform in azimuth–elevation space, given by

$$p(\mathbf{z}_{i,\ell,m}|\boldsymbol{\emptyset}) = \frac{1}{V_i} = \frac{1}{W_i^\theta W_i^\phi}, \quad \forall\, \ell, m, \tag{20}$$

where V_i is the total volume of the measurement space of sensor i.

Under the ML-PMHT framework, the term in the numerator on the right-hand side of (19) is given by

$$p(\mathbf{z}_{i,\ell,m}|\boldsymbol{\mathcal{X}}^k) = \pi_{i,0,\ell}\, p(\mathbf{z}_{i,\ell,m}|\emptyset) + \sum_{j=1}^{N^{\mathrm{t}}} \pi_{i,j,\ell}\, p(\mathbf{z}_{i,\ell,m}|\mathbf{X}_{j,\ell}), \tag{21}$$

which is a convex combination of the likelihoods based on the different possible measurement generating processes with the coefficients being the prior probabilities in (17). The term $\mathbf{X}_{j,\ell}$ is the global position of target j at time step ℓ and is given in (1). The relation between $\mathbf{X}_{j,\ell}$ and the batch joint target motion vector $\boldsymbol{\mathcal{X}}^k$ is given by (1) and (2).

Under the Gaussian measurement error assumption, the term $p(\mathbf{z}_{i,\ell,m}|\mathbf{X}_{j,\ell})$ is given by the multivariate Gaussian density

$$p(\mathbf{z}_{i,\ell,m}|\mathbf{X}_{j,\ell}) = \frac{1}{\sqrt{|2\pi\mathbf{R}_{i,\ell}|}} \mathrm{e}^{-0.5(\mathbf{z}_{i,\ell,m}-\bar{\mathbf{z}}_{i,j,\ell})^{\mathrm{T}}\mathbf{R}_{i,\ell}^{-1}(\mathbf{z}_{i,\ell,m}-\bar{\mathbf{z}}_{i,j,\ell})}. \tag{22}$$

Under the azimuth–elevation measurement model, $\bar{\mathbf{z}}_{i,j,\ell}$—the predicted measurement originating from target j at time step ℓ from sensor i—is given by (9). The sensor reference frame target position $\mathbf{x}_{i,j,\ell}$ required in the measurement prediction is given in terms of the global target position $\mathbf{X}_{j,\ell}$ by (14).

Combining (19)–(22) gives a final expression for the LLR in (23).

$$\mathcal{L}(\boldsymbol{\mathcal{X}}^k; \boldsymbol{\mathcal{Z}}^k) = \sum_{i=1}^{N^{\mathrm{s}}} \sum_{\ell=k-T^{\mathrm{b}}}^{k} \sum_{m=1}^{N_{i,\ell}^{\mathrm{z}}} \ln\left(\pi_{i,0,\ell} + V_i C_{i,\ell} \sum_{j=1}^{N^{\mathrm{t}}} \pi_{i,j,\ell}\, \mathrm{e}^{-0.5(\mathbf{z}_{i,\ell,m}-\bar{\mathbf{z}}_{i,j,\ell})^{\mathrm{T}}\mathbf{R}_{i,\ell}^{-1}(\mathbf{z}_{i,\ell,m}-\bar{\mathbf{z}}_{i,j,\ell})} \right). \tag{23}$$

In (23), $C_{i,\ell}$ is the Gaussian constant given by

$$C_{i,\ell} = \frac{1}{\sqrt{|2\pi\mathbf{R}_{i,\ell}|}}. \tag{24}$$

Since the remainder of this work deals with maximizing the LLR over the target motion space given a batch of measurements at time step k, the function in (23) will be denoted as $\mathcal{L}(\boldsymbol{\mathcal{X}}^k)$ for simplicity.

A. Choice of Priors

The performance of the ML-PMHT has been shown in pioneering works to be rather robust to changes in the prior probabilities in (17), (21), and (23). Qualitatively speaking, if it is expected that most of the measurements in any one scan will be clutter, the prior for the clutter process should be significantly higher than the priors for targets.

The simplest option is to naively set the clutter prior based on expected values. If $\lambda_{i,\ell}$ is the expected number of clutter measurements from sensor i at time ℓ and $\varphi_{i,j,\ell}$ is the expected number of measurements originating from target j, the prior for the clutter process can reasonably be set as

$$\pi_{i,0,\ell} = \frac{\lambda_{i,\ell}}{\lambda_{i,\ell} + \sum_j \varphi_{i,j,\ell}}, \tag{25}$$

and the prior for target j as

$$\pi_{i,j,\ell} = \frac{\varphi_{i,j,\ell}}{\lambda_{i,\ell} + \sum_j \varphi_{i,j,\ell}}. \tag{26}$$

The priors also provide a convenient way of working with sensors with restricted fields of view. If, for instance, a target position $\mathbf{x}_{i,j,\ell}$ in the reference frame of sensor i is such that either of the corresponding line-of-sight angles is out of the sensor's angular range (introduced in Section II.A to model a restricted field of view), then the corresponding "predicted observation" is $\bar{\mathbf{z}}_{i,j,k} = \emptyset$. That is, target j is expected to be out of view of sensor i at time ℓ. A convenient way to deal with this is to simply set $\pi_{i,j,\ell} = 0$, and adjust the other priors so that $\sum_j \pi_{i,j,\ell} = 1$.

IV. LIKELIHOOD MAXIMIZATION

As stated earlier in (3), the ML batch estimate up to time k is the $\boldsymbol{\mathcal{X}}^k$ that maximizes $\mathcal{L}(\boldsymbol{\mathcal{X}}^k)$. Since $\mathcal{L}(\boldsymbol{\mathcal{X}}^k)$ is highly non-linear, an analytic solution is not obtainable. However, given a batch of measurements and some fixed $\boldsymbol{\mathcal{X}}_0^k$, it is easy enough to evaluate $\mathcal{L}(\boldsymbol{\mathcal{X}}_0^k)$ using (9), (14), and (23). Therefore, maximization can be done via a hill climbing algorithm combined with other techniques (discussed later) to get within the neighborhood of the global maximum.

Generally speaking, any hill climbing algorithm functions by stepping around a parameter space attempting to find the global maximum in some function of the parameters. Obviously, the hill climber must be started at some initial point in the parameter space. For some applications, it would be perfectly reasonable to sample the initial point from a uniform distribution on the parameter space. In other applications, the measurement space and the parameter space are one and the same. In such cases, one may simply treat some observed measurement as the initial step in the hill climber. There are two main challenges in the likelihood maximization in this work. First, the measurement spaces are not the same as the parameter space. Second, due to clutter, there is not a single hill in the LLR (see Fig. 4). Instead, there are multiple "false" hills (local maxima) along with a single true hill (global maximum). In general, the true hill will be taller than the false hills. However, from the point of view of the hill climber, there is no way to determine how tall a given hill is at the start of climbing. This means that the climber can get stuck climbing the wrong hill. Thus, it would be to our advantage to have a method of initializing the climber as close to the true hill peak as pos-

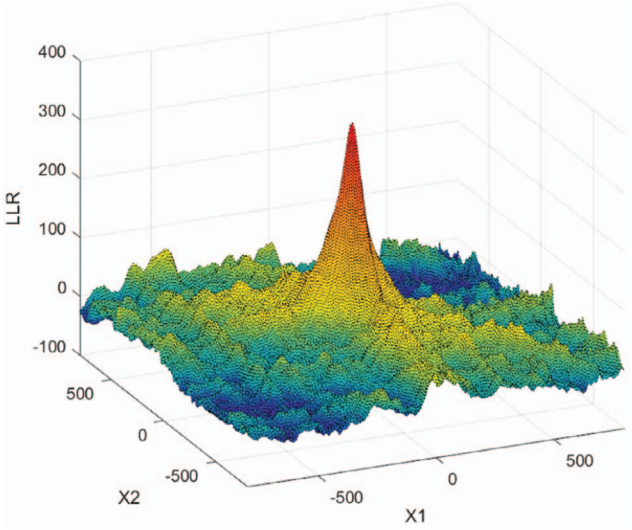


Fig. 4. The LLR surface centered on truth with two out of six dimensions varied (the global X coordinate of both the start and end point of the one target). The true hill (global max) is significantly taller than the rest (local maxima). But since hill climber termination is based on hill slope, the climber could potentially terminate at the top of a false hill. This demonstrates why it is important to initialize the climber as close to the true hill as possible.

sible. This routine will be referred to as "hill finding" in this work. However, "hill finding" is somewhat of a misnomer since the LLR surface has many hills. Precisely speaking, the routine is an attempt to get close (in the parameter space) to the one hill whose peak is the global maximum; the term "hill finding" is used for concision.

A. Hill Finding

The parameter space in this application is described by (2). For a scenario with N^{t} targets, the parameter exists in a $6N^{\mathrm{t}}$ dimensional space. Each six-dimensional sub-space parameterizes a line segment in the three-dimensional global tracking space—three dimensions for the start point and three for the end point. The line segments parameterized in this space will represent the estimate of the corresponding target's trajectory over the course of a batch. With this formulation, a "point" in the parameter space represents a group of N^{t} line segments in the tracking space. Maximizing the likelihood over this parameter space amounts to finding the group of line segments that best represents the target trajectories based the batch of measurements.

"Good" initialization of the hill climber involves setting the initial parameter vector to represent a "good guess" as to the target trajectory segment(s) during the batch. Given some prior knowledge of the targets' behavior, it is possible to predict a parameter to use for initializing the climber. However, for the sake of robustness and to deal with scenarios where no prior information about the targets is available, we have developed a data-driven method of initializing the hill finder. This method, detailed in the rest of this section, obtains a parameter initialization based only on the currently observed batch of measurements. Thus, the tracker can be started "blind"—with no prior information about the targets—and the data-driven hill finding should cause the tracker to converge on the target track(s) within a few batches. Based on the formulation of the parameter space, the ultimate goal of the hill finder—presented in the following, somewhat verbose discussion—is to obtain estimates of target locations in the three-dimensional tracking space at the start and end times of the batch of measurements.

Given the pose of sensor i at time step ℓ, $\{\mathbf{\Gamma}_{i,\ell},\ \boldsymbol{\xi}_{i,\ell}\}$, and a single two-dimensional measurement $\mathbf{z}_{i,\ell,m}$ from that sensor and time, there is not enough information present to solve for a potential target location in three-dimensional global space. There is, however, enough information to define a ray in three-dimensional global space that starts at the origin of the sensor coordinate system and extends infinitely in the line-of-sight direction indicated by the measurement. The azimuth–elevation measurement model is particularly convenient here since it directly gives the azimuth and elevation of this ray in the sensor reference frame.

Now since the algorithm is working with a group of at least two sensors, consider a pair of sensors $\{a, b\}$ at time ℓ and a pair of measurements $\{\mathbf{z}_{a,\ell},\ \mathbf{z}_{b,\ell}\}$—one from each sensor. Then consider the two corresponding line-of-sight rays, one starting at the origin of sensor a extending in the direction indicated by $\mathbf{z}_{a,\ell}$ and the other starting at the origin of sensor b extending in the direction indicated by $\mathbf{z}_{b,\ell}$. In the absence of clutter and measurement error, and given that the two measurements originated from the same target, this pair of rays would provide the precise location of the target as discussed in (for example) [1] and [8] by finding the point where the rays intersect. With additive measurement errors, however, these rays are unlikely to intersect. Furthermore, if one or both of the measurements are clutter originated, or if they do not originate from the same target, then the measurements do not have any meaning when considered as a pair. However, if the measurements in the pair both happen to originate from a target at a point $\mathbf{p}$ in the global space, then the rays indicated by the two measurements should "closely agree" on a point near $\mathbf{p}$, though they will not have an exact intersect due to the measurement errors.

Recall that at sample time ℓ, a single sensor i returns a "scan"—a set of measurements $\mathbf{Z}_{i,\ell}$ that includes all clutter-originated and target-originated measurements. So for each possible pair of sensors, the hill finder routine should check for "hit points" in three-dimensional space that both sensors "closely agree" upon. Obviously not every pair of measurements will "closely agree" on a point since many measurements are clutter-generated and it is not guaranteed that the two sensors are even "looking at" any of the same points (the sensors could, for instance, be placed back-to-back and faced in opposite directions). Thus, in a process of elimination, measurement pairs are subjected to a series of increasingly

strict tests. First of all, the tests must eliminate from consideration pairs of measurements that could not possibly correspond to a single target location in three-dimensional space. Then, pairs that are likely to be unrelated (one or both are clutter originated or they originate from different targets) must be eliminated. And, finally, any measurement pairs that remain must be condensed into composite point measurements in the three-dimensional tracking space—a point $\hat{\mathbf{p}}$ upon which they agree according to some criterion—and the "strength" of agreement should be quantified so that the composite measurements can be objectively ranked in quality.

1) Formation of Composite Point Measurements: This series of tests is presented next for a single measurement pair. The first level of tests are based on computing the closest approach between the pair of line-of-sight rays. This is heuristic but useful for computationally cheap elimination of measurement pairs that are most likely unrelated, which is especially useful in scenarios with large amounts of clutter in each scan. The final, more strict test is based on the iterative least-squares (ILS) estimator.

To formalize, consider a particular pair of sensors $\{a, b\}$, $a \neq b$, at a particular time ℓ. Take the pair of scans $\{\mathbf{Z}_{a,\ell},\ \mathbf{Z}_{b,\ell}\}$, and let

$$N^{z}_{i,\ell} = |\mathbf{Z}_{i,\ell}|, \qquad i = a, b, \tag{27}$$

be the number of individual measurements in the scans.

Now consider some pair of measurements (one from each scan)

$$\{\mathbf{z}_{a,\ell,m},\ \mathbf{z}_{b,\ell,n}\}, \quad m \in \{1, \ldots, N^{z}_{a,\ell}\},\ n \in \{1, \ldots, N^{z}_{b,\ell}\}. \tag{28}$$

Let us momentarily drop the time (ℓ) and measurement pair (m, n) indexing and simply consider a particular measurement from sensor a, call it $\mathbf{z}_a$, a particular measurement from sensor b, call it $\mathbf{z}_b$, with both measurements taken at the same time. Each measurement can be taken to represent a ray (half line). So the pair of measurements yields a pair of rays parameterized by a pair of origin points and a pair of unit vectors that indicate the rays' pointing directions *in the global coordinate system*.

The origin of the first ray is $\boldsymbol{\xi}_a$—the location of sensor a, which is assumed to be known. Similarly, the second ray has origin point $\boldsymbol{\xi}_b$. The direction vectors are found by first obtaining the unit vectors in their respective sensor reference frames and transforming them both into the global reference frame with the known sensor orientations. In the case of azimuth–elevation measurements, the unit direction vectors are given by

$$\mathbf{v}_i = \boldsymbol{\Gamma}_i \begin{bmatrix} \sin\phi_i \\ \sin\theta_i \cos\phi_i \\ \cos\theta_i \cos\phi_i \end{bmatrix} \qquad i = a, b, \tag{29}$$

where θ_i, ϕ_i are the individual components of $\mathbf{z}_i$ and $\boldsymbol{\Gamma}_i$ is the rotation matrix of the ith sensor pose.

For any non-parallel pair of rays in three-dimensional space, there is a single line segment somewhere that connects the rays and is perpendicular to both rays. The end points of this line segment are the points of closest approach of the rays, and the length is the minimum distance between the rays [19]. For this case, solve for the end points of the minimum distance segment with

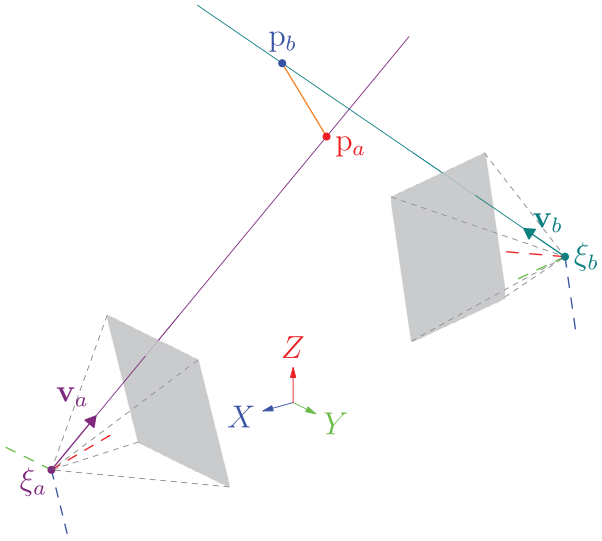


Fig. 5. Notional illustration of the points of closest approach (red and blue points) between two line-of-sight measurements.

$$\mathbf{p}^{\mathrm{md}}_i = \rho_i \mathbf{v}_i + \boldsymbol{\xi}_i, \quad i = a, b, \tag{30}$$

where $\mathbf{p}^{\mathrm{md}}_i$ is the location of the closest approach that lies on ray i. The scalar values ρ_a, ρ_b determine the distance along each ray at which the closest approach occurs. Let

$$\mathbf{c} \triangleq \boldsymbol{\xi}_b - \boldsymbol{\xi}_a, \tag{31}$$

then these scalar values are given by

$$\rho_a = \frac{-(\mathbf{v}_a \cdot \mathbf{v}_b)(\mathbf{v}_b \cdot \mathbf{c}) + (\mathbf{v}_a \cdot \mathbf{c})}{1 - (\mathbf{v}_a \cdot \mathbf{v}_b)^2}, \tag{32}$$

$$\rho_b = \frac{(\mathbf{v}_a \cdot \mathbf{v}_b)(\mathbf{v}_a \cdot \mathbf{c}) - (\mathbf{v}_b \cdot \mathbf{c})}{1 - (\mathbf{v}_a \cdot \mathbf{v}_b)^2}, \tag{33}$$

where the dot indicates a vector dot product. Both values are defined as long as the two measurement rays are not perfectly parallel, which happens with probability zero. Refer to Fig. 5 for an illustration of the closest approach between an example pair of measurements.

First, a test can be performed by considering just the signs of the scalar values ρ_a, ρ_b. These are the Cartesian distances along the rays where the closest approach points occur. Thus, if either ρ_a or ρ_b is negative, it means the corresponding closest approach point occurs "behind the sensor." (It is assumed that the negative z half of the sensor coordinate frame is never observable.) So, if either value is negative, reject the corresponding pair of measurements as being indicative of a "hit" on a target.

Furthermore, in some settings, it would be reasonable to set maximum limits for the values ρ_a, ρ_b such that if either value exceeds its maximum, the pair of

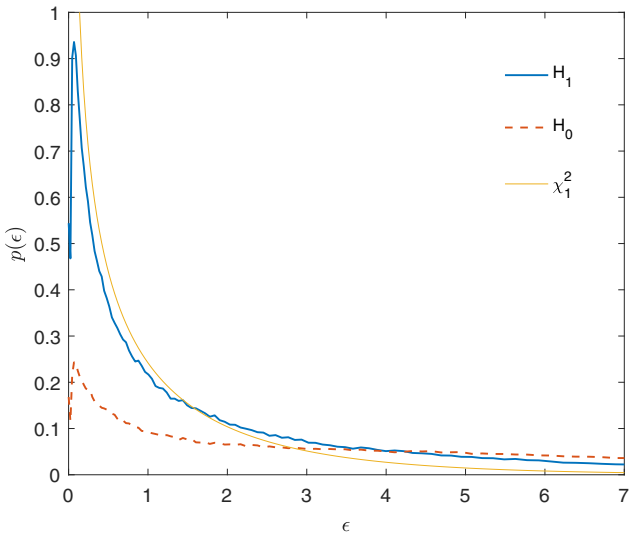


Fig. 6. The empirical distributions of the normalized measurement error squared under the binary hypotheses: H_0: the measurements used in the ILS estimate are unrelated and H_1: the measurements used in the ILS estimate originated from the same target. The one-DOF Chi-squared distribution is also plotted for reference. The discrepancy between the theoretical Chi-squared distribution and empirical distribution under H_1 is due to the thresholded termination of the ILS. The distributions would match if the ILS was run to exact termination with perfect numerical precision.

measurements is eliminated. For instance, if visibility is such that the user knows that no sensor can see farther than 10 000 m, and two measurement rays have a closest approach point that is 15 000 m away from one of the sensors, then that pair of measurements could also reasonably be rejected as originating from a common target. This would also be of use if it is known that the target tracking space down range of one or both sensors has a hard limit—e.g., the sensors are orbiting the earth at some known altitude and are pointed toward the earth's surface.

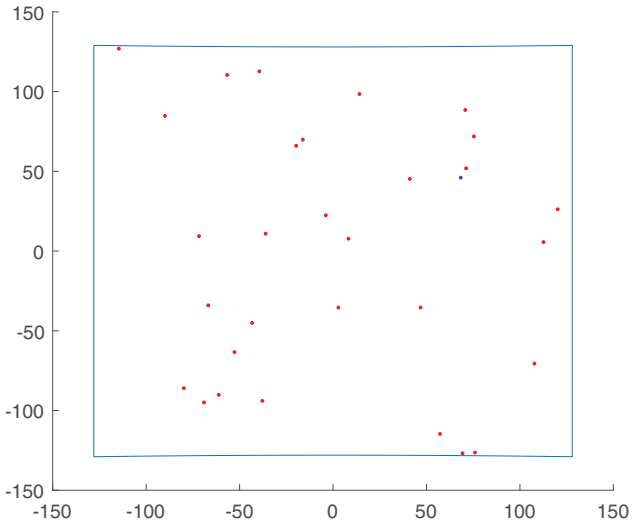


Fig. 7. Example camera view of hits for a single scan for a clutter level of $\lambda = 30$.

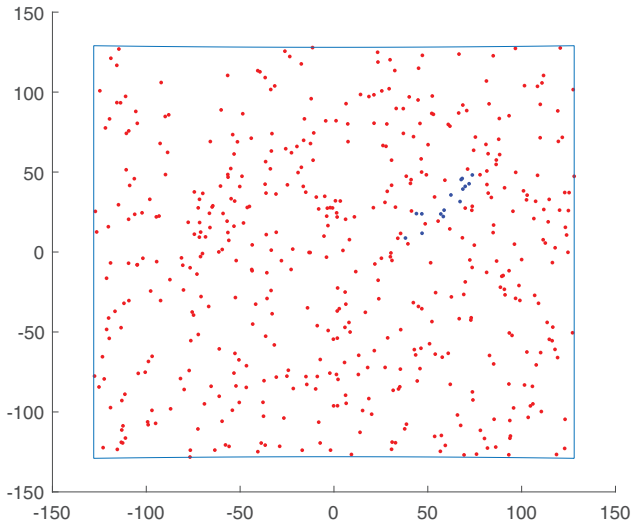


Fig. 8. Example camera view of superimposed scans for an entire batch for a clutter level of $\lambda = 30$. Blue dots indicate true target hits and red dots are clutter. Note that without the color coding, it is not clearly evident to the human eye where the target track is located. The blue outline shows the image edge that results from setting limits on the azimuth and elevation angles.

A final heuristic test involves setting a threshold τ_{md} on the length of the minimum distance segment such that if

$$||\mathbf{p}_a^{\text{md}} - \mathbf{p}_b^{\text{md}}||_2 \geq \tau_{\text{md}}, \tag{34}$$

then the measurement pair is eliminated from consideration. The ideal value for τ_{md} depends on the particular scenario and the desired level of restraint in eliminating measurement pairs. Qualitatively speaking, the farther down range the targets are expected to appear, the greater the τ_{md} should be. And, if it is found that the algorithm is considering more measurement pairs than the user finds reasonable, then τ_{md} can be decreased. This is perhaps the most heuristic of the series of tests. However, in simulated scenarios with large amounts of clutter, and sensors, which are known to be observing a common space in which the targets are known to exist, it has been found to be the workhorse test that eliminates all but the most likely-to-be-related measurement pairs.

The tests presented thus far serve to eliminate line-of-sight measurement pairs that either do not point toward a common space, point at a space that is physically too far away for the sensors to observe, point at a space where targets are not likely to exist, or are likely unrelated based on not approaching each other within a reasonably constrained space. Now, any measurement pairs that remain must be either eliminated as well or consolidated into a single composite measurement point in three-dimensional space. Suppose that the measurement pair under consideration—$\{\mathbf{z}_a, \mathbf{z}_b\}$—has passed the simple tests involving the closest approach between the lines-of-sight. If those tests were reasonably well-tuned to the operating scenario, the fact that the measurement pair has passed increases the likelihood that both

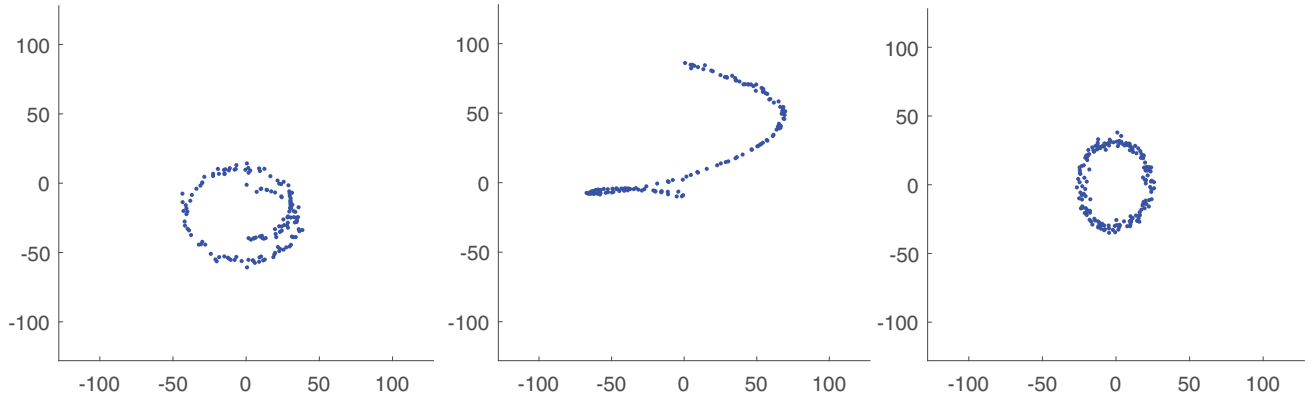


Fig. 9. The true hits received by each camera over the course of the entire scenario. Missed detections are accounted for (not visible).

measurements in the pair originated from a common target. Thus, a composite point measurement in the three-dimensional tracking space based on the pair of measurements is likely to be "meaningful"—an estimate of a true target position.

The ILS estimator provides a mathematically rigorous method to form a composite point measurement based on the pair of line-of-sight measurements. It functions by iteratively solving for the point $\hat{\mathbf{p}}$ in the three-dimensional tracking space, which minimizes the normalized measurement error squared implied by the measurement pair due to a target at point $\hat{\mathbf{p}}$, which is a scalar value given by

$$\epsilon = \sum_{i=a,b} (\mathbf{z}_i - \bar{\mathbf{z}}_i(\hat{\mathbf{p}}))' \mathbf{R}_i^{-1} (\mathbf{z}_i - \bar{\mathbf{z}}_i(\hat{\mathbf{p}})). \tag{35}$$

where $(\cdot)'$ denotes the matrix transpose. The term $\bar{\mathbf{z}}_i(\hat{\mathbf{p}})$ is the zero-error measurement returned by sensor i due to a target at $\hat{\mathbf{p}}$. Notice the implicit assumption that both of the measurements in the pair have in fact originated from the same target—something that cannot be known for certain in this case. However, due to the previous series of tests based on the closest approach points, the ILS estimate will be calculated only for likely-to-be-related measurement pairs. Furthermore, the final minimized value of ϵ will be used as a final test statistic to eliminate all but the strongest composite measurements.

The ILS estimator using line-of-sight measurements in azimuth–elevation form is given in [13], and used for related work in [9]. It is summarized here. First, form a 4×1 vector by stacking the pair of measurement column vectors

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_a \\ \mathbf{z}_b \end{bmatrix} \tag{36}$$

and form the corresponding 4×4 covariance matrix

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_b \end{bmatrix}, \tag{37}$$

where $\mathbf{0}$ is a 2×2 matrix of zeros. Then $\hat{\mathbf{p}}_q$—the ILS estimate at the qth iteration—is updated with an additive term as

$$\hat{\mathbf{p}}_{q+1} = \hat{\mathbf{p}}_q + \boldsymbol{\Delta}_q, \tag{38}$$

where the additive update term is calculated as

$$\boldsymbol{\Delta}_q = \left[\mathbf{H}_q' \mathbf{R}^{-1} \mathbf{H}_q \right]^{-1} \mathbf{H}_q' \mathbf{R}^{-1} \left[\mathbf{z} - \bar{\mathbf{z}}(\hat{\mathbf{p}}_q) \right], \tag{39}$$

where the 4×1 zero-error measurement vector is given by stacking the individual zero-error measurement vectors as

$$\bar{\mathbf{z}}(\hat{\mathbf{p}}_q) = \begin{bmatrix} \bar{\mathbf{z}}_a(\hat{\mathbf{p}}_q) \\ \bar{\mathbf{z}}_b(\hat{\mathbf{p}}_q) \end{bmatrix}, \tag{40}$$

in which the individual vectors $\bar{\mathbf{z}}_i(\hat{\mathbf{p}}_q)$ are obtained by first setting $\mathbf{X}_{j,\ell} = \hat{\mathbf{p}}_q$ in (14), and then substituting the result into (9). The matrix $\mathbf{H}_q$, defined as

$$\mathbf{H}_q = \left. \frac{\partial \bar{\mathbf{z}}(\mathbf{X})}{\partial \mathbf{X}} \right|_{\mathbf{X}=\hat{\mathbf{p}}_q}, \tag{41}$$

is the 4×3 Jacobian matrix of the stacked zero-error measurement vector with respect to Cartesian position in the *global reference frame*, evaluated at the current ILS estimate.

The formulas for the individual elements of the Jacobian matrix and the initialization of the ILS estimator are given in the Appendix.

The ILS estimator is terminated by setting a threshold τ_Δ and iterating until

$$||\boldsymbol{\Delta}_q||_2 \leq \tau_\Delta \tag{42}$$

and recording the final estimate as

$$\hat{\mathbf{p}} = \hat{\mathbf{p}}_{q+1}. \tag{43}$$

Once the final ILS estimate is obtained, the minimum normalized measurement error squared given by (35) is theoretically Chi-squared distributed with one DOF *given that the pair of measurements used in the estimate originated from a common target* [4]. If the pair of measurements used in the estimate is unrelated despite having passed the previous tests, the distribution of the minimum normalized measurement error has no known closed form. However, the distribution is obtained empirically through simulation and plotted in Fig. 6. The

shape of this distribution is much like the Chi-squared one DOF distribution, but with a much heavier tail. This allows for a final thresholded test using ϵ—calculated with the formula in (35)—as the test statistic. That is, if the value of ϵ resulting from the ILS estimate based on the measurement pair under consideration is above some threshold τ_ϵ, then the pair and the resulting composite measurement can be eliminated from consideration. Otherwise, the composite measurement has passed all elimination tests and is taken to correspond to an actual target. A composite measurement that passes this final test can be assigned a quantitative score given by

$$\sigma = 1 - \chi_1^2(\epsilon) \tag{44}$$

—the complement of the one DOF Chi-squared cumulative distribution function (cdf) evaluated at the minimized normalized measurement error squared. This will provide a score in the range $[0, 1]$, with better estimates receiving higher scores.

The elimination tests above were presented in terms of a single pair of measurements from a particular pair of sensors (a, b) at a single time step, and the indexes of the measurement pair (m, n) and time step (ℓ) were omitted throughout. Now, suppose that the algorithm performs the entire series of elimination tests for every element of the set

$$\{\{\mathbf{z}_{a,\ell,m},\ \mathbf{z}_{b,\ell,n}\}\ :\ 1 \le m \le N^{\mathrm{z}}_{a,\ell} \quad 1 \le n \le N^{\mathrm{z}}_{b,\ell}\}$$

—every possible pair of measurements from the scans of sensors a and b at time ℓ. Record to memory all the resulting composite measurement points that have passed all the elimination tests along with their corresponding Chi-squared scores given by (44) as the set of parameter pairs

$$\tilde{\mathbf{P}}_{(a,b),\ell} = \left\{(\hat{\mathbf{p}}, \sigma)_p\right\}_{p=1}^{N^{\mathrm{p}}_{(a,b),\ell}}, \tag{45}$$

where $N^{\mathrm{p}}_{(a,b),\ell}$ is the number of composite measurements from sensor pair (a, b) at time step ℓ that have not been eliminated.

Notice that along with the obvious dependence on time index ℓ, $\tilde{\mathbf{P}}_{(a,b),\ell}$ is also dependent on the *pair* (a, b). That is, the list of points obtained at a particular time depends on which pair of sensors is being used to look for points. With a group of $N^{\mathrm{t}} > 2$ sensors, find the list of hit points from each possible pair of sensors and form the total list

$$\mathcal{P}_\ell = \bigcup_{(a,b)\in\mathbf{r}} \tilde{\mathbf{P}}_{(a,b),\ell}, \tag{46}$$

where $\mathbf{r}$ is the set of all unordered pairs of the integers $[1, N^{\mathrm{s}}]$.

2) Formation of Initial Parameter Estimates: To form initial parameter vectors to pass to the climber at estimation time step k, suppose that $\mathcal{P}_\ell$ has been obtained for every time step $\ell \le k$. Potential target track segments over the current batch are formed by pairing composite measurements from the current batch start time in $\mathcal{P}_{k-N^{\mathrm{b}}+1}$ with composite measurements from the current batch "leading edge" time in $\mathcal{P}_k$, which parameterizes line segments in global three-dimensional space. If the user has knowledge of a minimum and/or maximum possible target velocity (in units length/sampling period), then it can be used here to select only the reasonable potential track segments. The segments are also assigned a score equal to the product of the Chi-squared score of the composite measurements that make up its end points. This is useful in quantitatively ranking which segments are the best if the number of segments sent to the hill climber must be limited.

Formally, let $s = k - N^{\mathrm{b}} + 1$ be the batch start time step, and form the set of parameter pairs given in (47),

$$\begin{aligned}\mathcal{Q}^k = \Bigg\{ \Bigg(\mathbf{Q} = \begin{bmatrix} \hat{\mathbf{p}}_{s,p} \\ \hat{\mathbf{p}}_{k,q} \end{bmatrix},\ \delta = \sigma_{s,p}\sigma_{k,q} \Bigg) \\ :\ v_{\min} T^{\mathrm{b}} \le ||\hat{\mathbf{p}}_{s,p} - \hat{\mathbf{p}}_{k,q}||_2 \le v_{\max} T^{\mathrm{b}}, \\ 1 \le p \le |\mathcal{P}_s|, \quad 1 \le q \le |\mathcal{P}_k| \Bigg\},\end{aligned} \tag{47}$$

where $\hat{\mathbf{p}}_{s,p}$ is the pth composite measurement in the set of composite measurements from the batch start time step s, and $\sigma_{s,p}$ is the measurement's Chi-squared score. Similarly, $\hat{\mathbf{p}}_{k,q}$, $\sigma_{k,q}$ are the individual members of the qth element in the set of composite measurements from the batch leading edge time step k. Let $[v_{\min},\ v_{\max}]$ represent the range of possible target velocities and $T^{\mathrm{b}} = N^{\mathrm{b}} - 1$ be the *duration* of the batch in sampling periods. Forgive the reuse of the index q—it was used in Section IV.A.1 for an unrelated purpose.

To be verbose, each element of the set $\mathcal{Q}^k$ contains a 6×1 column vector parameterizing a line segment in three-dimensional space that satisfies a length restriction, paired with a scoring value that represents the "target indication strength" of the segment based on the Chi-squared scores of the two composite point measurements that parameterize the segment. Notice that only composite measurements from the start and end time steps of the batch are used to populate $\mathcal{Q}^k$. If the user finds that this does not provide enough target track segment estimates, then pairs of composite measurements from intermediate batch times can be used to form segments that are projected to the start and end batch times. That is, select pairs of composite measurements, one from time step t and one from time step ℓ, such that $s \le t < \ell \le k$ and—based on the assumption of a constant sampling period—compute the individual 3×1 components of the stacked vector $\mathbf{Q}$ in (47) as

$$\hat{\mathbf{p}}_s = \hat{\mathbf{p}}_\ell + \left(\frac{\ell - s}{\ell - t}\right)(\hat{\mathbf{p}}_t - \hat{\mathbf{p}}_\ell), \tag{48}$$

$$\hat{\mathbf{p}}_k = \hat{\mathbf{p}}_t + \left(\frac{k - t}{\ell - t}\right)(\hat{\mathbf{p}}_\ell - \hat{\mathbf{p}}_t), \tag{49}$$

and assign the projected segment a score equal to the product of the Chi-squared scores of the two composite measurements used in the calculation. This segment projection is particularly useful in scenarios where all sensors have a low probability of target detection in any one scan since, for there to exist a meaningful composite measurement at any one time step, at least two sensors will need to have detected the same target at that sampling instant.

The likelihood maximization will assume the number of targets based on the size of the initial parameter vector it is passed. In order to choose a good initial vector that parameterizes N^{t} targets, sample N^{t} segments from $\mathcal{Q}^k$ *without replacement*, and stack the individual segment vectors into a $6N^{\mathrm{t}} \times 1$ parameter vector of the form given in (2). The sampling should give preference to segments with high δ scores. A decent way to accomplish this is to use a high-level function like MATLAB's datasample() and sample element r of $\mathcal{Q}^k$ with the weight of δ_r relative to all other δ values. If the number of segments in $\mathcal{Q}^k$ is small enough that it will not be computationally prohibitive to simply run a new instance of the hill climber for every possible initial parameter vector, then this sampling can be avoided altogether.

Thus far, this section has discussed a method of using observed data to initialize the hill climber. This has been found to perform well enough as the sole method of hill climber initialization at the beginning of each batch. However, performance can be improved further if the target motion for the current batch is predicted from the best previous batch estimate (under the constant velocity assumption). The predicted segments along with segments sampled from $\mathcal{Q}^k$ are then each used to initialize individual instances of the hill climber. In simulation, it was found that this prediction aids the tracker in "sticking" to the track once it has a good estimate. On the other hand, it cannot be used as the only method of initializing the hill climber since the data-driven hill finder is needed to converge on a good estimate in the first place and to recover if the track is ever lost.

B. Hill Climbing

There are many types and variations of hill climbers. The maximization in this work uses the conjugate gradient method implemented in Python's SciPy "optimize" library [24]. It should be pointed out that an elegant expectation-maximization (EM) approach could also be used (see [2]).

In some cases, the user might be interested in deciding whether there is even a single target present or not (target detection). In this case, one could pass the hill climber an initial vector $\mathcal{X}_0^k$ that implies a single target (six elements), and allow the climber to run till it reaches a peak. Then, compare the value of the LLR at this peak to some threshold to decide if there is a target present. That is, if the peak of hill is below some height, decide that the hill is just due to a randomly occurring pattern in clutter instead of an actual target. The challenge with this is picking a good threshold for this test. The peak height of a given LLR hill depends heavily on the number of sensors being used and the geometric arrangement of the sensors relative to each other and the targets. The simulated detection performance is discussed in Section V.B.

C. Tracker Summary

Tracking is performed in a "sliding batch" fashion where after an estimate is obtained, the leading edge of the batch slides forward by some number of sampling periods less than the length of the batch. This means that consecutive batch estimates share some observations and are thus correlated. The following will give a step-by-step synopsis of the algorithm:

1) At time step k, take the N^{b} most recent scans from each of the N^{s} sensors to form the current measurement batch. That is, k indicates the leading edge of the batch.
2) For each time step in the batch, for each possible *pair* of sensors, obtain via the process of elimination in Section IV.A.1 pairs of measurements from the pair of sensors that "strongly agree" on some point in the global space. Record the resulting composite measurements.
3) With composite measurements obtained in the previous step, form line segments representing target tracks over the course of the batch. These segments are parameterized by their start and end points. They can be formed either by pairing hit points from the start and end times of the batch or by considering pairs of hit points from intermediate time steps and projecting out to the end points of the batch. If information is available about maximum and/or minimum target speed, this can be used to eliminate segments that are either too long or too short.
4) If working with a multi-target scenario with N^{t} targets, then form potential target parameters by combining line segments from the previous step into groups of size N^{t}. In a single target scenario, any one of the line segments can be taken as a potential target parameter.
5) (Optional) Form a predicted target parameter by using a constant velocity assumption to predict the target trajectories over the current batch based on the ML estimate from the previous batch.
6) For each of the parameter vectors obtained in 4) and 5), evaluate the LLR. Keep the N best parameters according the LLR value.
7) Initialize N hill climbers with each of the N best parameters from the previous step. Allow climbers to run until terminal condition or until some maximum number of steps has been exceeded. Take the param-

eter indicated by the climber that reached the highest peak in LLR to be the ML estimate for the current batch.

8) Increment k by the desired batch slide amount. Return to 1).

It is of interest to summarize the computational complexity of the method for a single batch estimate. As presented, the expected computational complexity of the hill finding is $\mathcal{O}\left((\lambda + p_{\mathrm{d}}N^{\mathrm{t}})^2 \frac{N^{\mathrm{s}}(N^{\mathrm{s}}-1)}{2}\right)$, where $\lambda + p_{\mathrm{d}}N^{\mathrm{t}}$ is the expected total number of measurements per scan (clutter plus targets) and $\frac{N^{\mathrm{s}}(N^{\mathrm{s}}-1)}{2}$ is the number of unique sensor pairs. That is, the point finding (optimization-initialization, or hill finding) cost is quadratic in both the number of measurements per scan and in the number of sensors. While the expected number of measurements is usually dictated by "nature," settings with many sensors may require care when choosing which sensor pairs are used to find points.

The evaluation of the likelihood in (23) has expected complexity $\mathcal{O}\left((\lambda + p_{\mathrm{d}}N^{\mathrm{t}})N^{\mathrm{s}}N^{\mathrm{t}}\right)$. The likelihood maximization requires approximation of the gradient of the objective likelihood function via two-point differencing, which requires $6N^{\mathrm{t}}+1$ evaluations of the objective function (one for each element of the parameter vector plus a reference evaluation). Thus, the overall likelihood maximization has complexity $\mathcal{O}\left(\left[\lambda(N^{\mathrm{t}})^2 + p_{\mathrm{d}}(N^{\mathrm{t}})^3\right]N^{\mathrm{s}}\right)$. The number of targets assumed by the optimization is the most significant factor in computational cost. While $(N^{\mathrm{t}})^3$ is the asymptotically dominant term, the entire expression $\lambda(N^{\mathrm{t}})^2 + p_{\mathrm{d}}(N^{\mathrm{t}})^3$ is noted since, in most practical settings, $\lambda \gg p_{\mathrm{d}}N^{\mathrm{t}}$ (there is usually much more clutter than target-originated measurements), so the clutter level can dominate practical computation. Both the hill finder and maximizer also have simple linear complexity in the batch length N^{b}.

V. RESULTS

The following sections present simulated results for the sake of testing the presented method. The detection performance (performance of a test to decide whether or not a target is present) is given in Section V.B. The tracking performance for a single target under various states of nature is studied in Section V.C. In Section V.D, a comparison is made to a decentralized method in which each sensor obtains an ML estimate of the track segment in its own measurement space, and then batch estimates from pairs of sensors are fused to obtain track segment estimates in the global space. This is in contrast to the method presented in the main body of this work could be considered a fuse-before-track method since the likelihood involves measurements from all sensors. Finally, a scenario with 2 targets is simulated, and a comparison is made between the tracking performance of the presented method and the JPDA method presented in [20].

A. Single Target Scenario Setup

The scenario has a single target and three sensors. The target travels along an upward spiraling path at a constant speed, see Fig. 9. Notice that, technically, the curvature of the path violates the constant velocity assumption in the target motion model. However, due to the constant speed of the target and the small amount of curvature that occurs over the course of any one batch, constant velocity is a sufficient approximation.

Each simulated sensor has a restricted, conical field of view with an angular range of 20°. The sensors are placed so that, for the majority of the target's trajectory, it is in view of all sensors. The sensor locations are all out of view in Fig. 11, but referring to the coordinate system in the figure:

- Sensor 1 is stationary at position (−18000,0,0), and aimed in the ^{+}X direction (toward the origin) and slightly up. This mimics a camera viewing the sky from the surface of the earth.
- Sensor 2 is stationary at position (0,0,60000), and aimed toward the origin. This gives an overhead view from a very high altitude.
- Sensor 3 is in motion. It orbits the Z axis at a height of 12 000 above the X–Y plane and with a radius of 10 000. Its orientation changes so that it is consistently aimed down and in toward the origin. Its orbital speed is such that it makes only half an orbit throughout the scenario.

In a real scenario, the expected number of clutter λ and the probability of target detection p_{d} are functions of SNR and the threshold used in the measurement extraction process. Qualitatively speaking, at a fixed SNR, lower extraction threshold values will result in more clutter but higher probability of target detection.

For simulation purposes, it is assumed that the measurement extraction process is Gaussian as in [11]. That is, assume some extraction threshold τ_{me}, which, when applied to a test for a measurement "hit" in some particular section of the sensor space where a target is not present, results in a single clutter measurement with a probability of "false alarm" given by

$$p_{\mathrm{f}} = \mathrm{Q}\left(\frac{\tau_{\mathrm{me}}}{\sigma}\right), \tag{50}$$

where $\mathrm{Q}(\cdot)$ is the Gaussian Q function and τ_{me} has been normalized by the measurement intensity standard deviation σ. Then, for a fixed SNR, the probability of target detection is

$$p_{\mathrm{d}} = \mathrm{Q}(\mathrm{Q}^{-1}(p_{\mathrm{f}}) - \mathrm{SNR}). \tag{51}$$

For the sake of simulation, it is assumed that there are a Poisson random number of clutter measurements in each scan with expected value λ, and the expected number is approximately related to the clutter level by

$$p_{\mathrm{f}} = \frac{\lambda}{N_{\mathrm{c}}}, \tag{52}$$

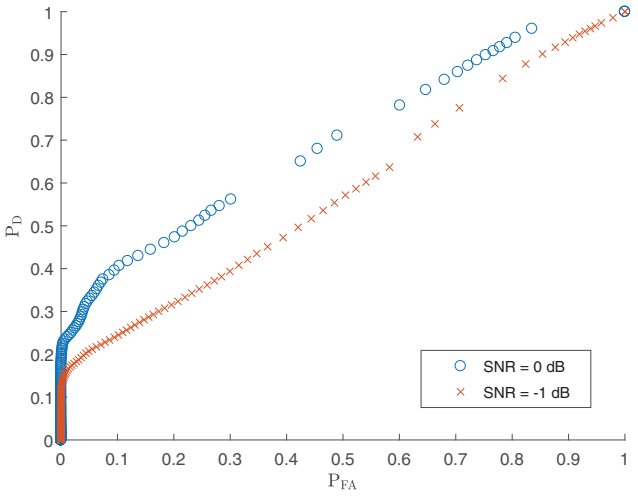


Fig. 10. The ROC curves for target detection based on the ML-PMHT likelihood at different SNR levels. The probability of target detection at individual sensors is fixed at $p_d = 0.45$. This implies a clutter level of $\lambda = \sim 54$ at an SNR of 0 dB and $\lambda = \sim 74$ at an SNR of -1 dB

where N_c represents some total number of resolved "cells" in the sensor space that are tested by the measurement extraction. It is assumed here that all sensors have $N_c = 400$.

The simulation uses azimuth–elevation measurements. Target-originated measurements have Gaussian additive error where the individual components are assumed independent and each has standard deviation $\sigma_\theta = \sigma_\phi = 0.25^\circ$. The small measurement error variance is used to mimic the behavior of cameras, which generally exhibit good measurement accuracy. The other "error-inducing" processes (clutter and missed detections) are considered more interesting in this context.

Unless otherwise noted, results use a fixed batch size $N^b = 18$, and the batch slides five sampling periods from one estimate to the next.

B. Detection Performance

It is of interest how well the algorithm does at detecting the presence of a target. Detection would be performed by choosing some threshold value and declaring a detection if the LLR for the current batch exceeds the threshold. To test the detection performance, the algorithm is run for 100 Monte Carlo runs under the null hypothesis—no target present. These data are combined with the data from the simulation with a target present to form the receiver operating characteristic (ROC) curves shown in Fig. 10. When the target is present, each sensor measures it with fixed probability of detection $p_d = 0.45$.

At first glance, the detection performance appears poor. However, it is important to note that detection is performed on a batch-to-batch basis. That is, if the detection was operating at a point on the ROC with $P_D = 0.3$ and $P_{FA} = 0.03$, then the user could expect to get a detection within four batches of the target appearing,

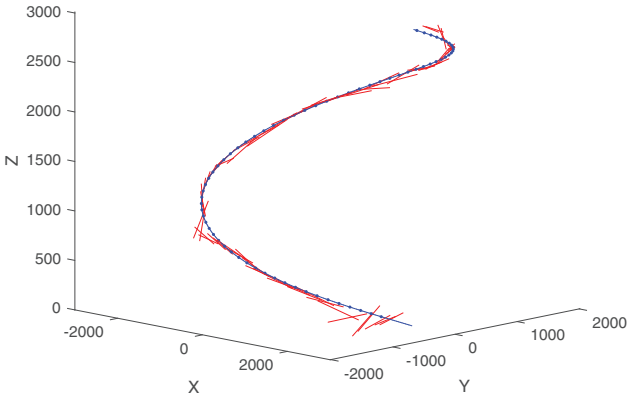


Fig. 11. The true target track (blue) plotted along with the batch estimates (red) for a single Monte Carlo run with $\lambda = 54$ and $p_d = 0.45$.

while expecting a false detection only once in every ~33 batches when there is no target present. When framed in this manner, the detection performance is acceptable. It is also important to emphasize that the detection and false alarm probabilities on the axes of Fig. 10 are not the same as the detection and false alarm probabilities in (50) and (51), which are properties of the underlying measurement extraction process.

C. Tracking Performance

When tracking with a single target present, the effect of the hill finder can be seen in the first few ML batch estimates in Fig. 11. While the first few estimates are not on track, the hill finder enables the tracker to converge on the track within a handful of batches. If instead the hill climbing was initialized randomly throughout the entire tracking space, it would have a tendency to settle on false hills instead of converging to the true track. There is a point about one third of the way into the scenario where the algorithm briefly does a poor job tracking the target. This is due to the geometry of the target relative to the sensors being less than ideal at that time—two sensors have almost anti-parallel lines-of-sight on the target. The resulting deviation in the estimate can be seen in Fig. 11 at the spot where the batch estimates drift away from the true track and in Fig. 12 by the spike in estimation error around time step 120. The spike in error is short-lived, however, since the hill finder compensates as soon as the sensors have good visibility on the target again.

We wish to also study the performance of the algorithm in terms of how often it is on track. To do so, we must first quantify what it means to be "on track." One way to define "on track" is to find the root mean square error (RMSE) over the course of each batch. Then, if the RMSE over the course of a particular batch is below some threshold, declare the algorithm to be on track for that batch. It is of interest to study the tracking performance based on the operating characteristic of the measurement extraction process. The SNR is fixed and sim-

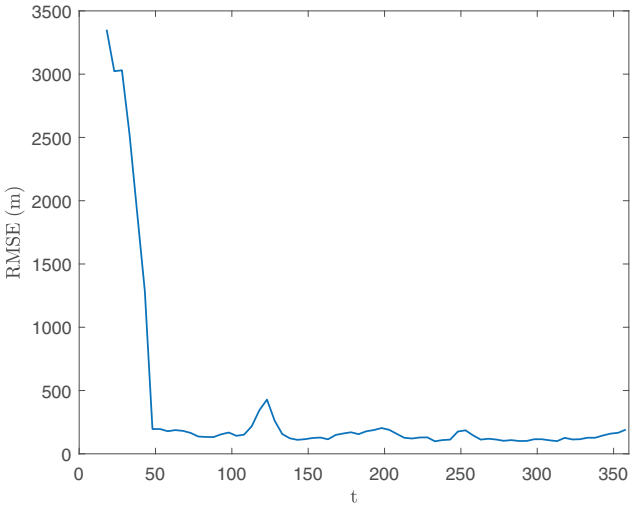


Fig. 12. The RMS distance error between the estimate and true target position versus time step. Averaged over 100 MC runs with $\lambda = 54$ and $p_{\mathrm{d}} = 0.45$.

ulations are performed over a range of normalized measurement extraction threshold values, which is equivalent to a range of clutter levels via (50) and (52). The empirical probability of the algorithm being on track (denoted P_{T}) is plotted for three different SNR values in Fig. 13. The RMSE threshold used to declare whether the algorithm is on track or not is set at 200 using the results in Fig. 12—slightly higher than the RMSE to which the algorithm empirically converges.

For each plot, a peak in tracking performance appears, above which the decreasing probability of target detection has a negative effect on the hill finder, and below which the increasing level of clutter results in decreasing estimation quality. The trend in the horizontal location of the peaks suggests that the lower the SNR of

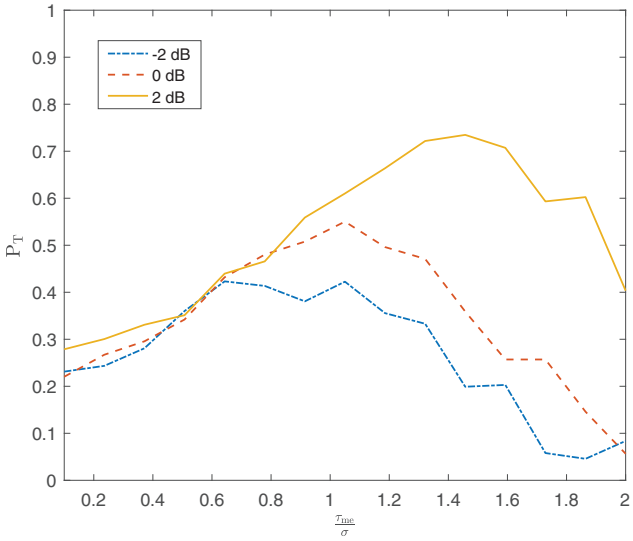


Fig. 13. The probability of being on track versus the normalized measurement extraction threshold. Different plots are for various measurement extraction SNR values. Results are averaged over 20 Monte Carlo runs of the scenario—or 1400 batches—for each data point.

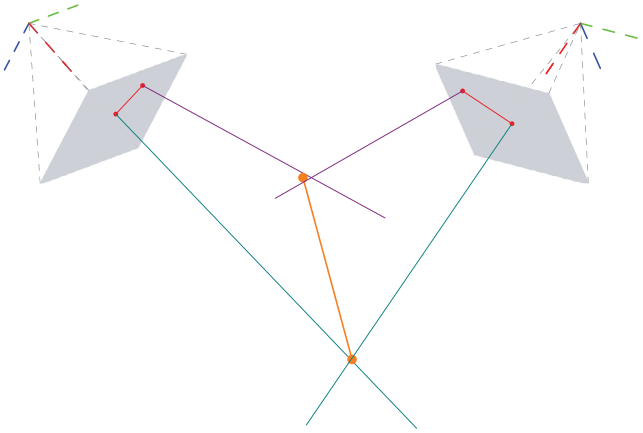


Fig. 14. Notional representation of fusing two batch estimates from two separate cameras (red segments) into a single estimate in global space (orange segment).

the tracking scenario, the lower the ideal measurement extraction threshold. That is, in scenarios with lower target visibility, it is better to operate in a "high clutter/high detection probability" regime. And when targets are more visible, it is ideal to compromise on the detection probability with the pay-off of having less clutter. The vertical location of the peaks are perhaps less informative since the probability of detection is defined in terms of an RMSE threshold (a peak would be higher if a higher error threshold was used for declaring the algorithm on track). However, when the same threshold is used for each data series in Fig. 13, the predictable trend of increasing peak performance with increasing SNR can be observed.

These results assume a Gaussian intensity detection structure for the measurement extraction, which represents a "worst case." Of course, if there is some other extraction process with which a higher detection probability and/or a lower level of clutter can be obtained (if, for instance, the preprocessing of images involved software that used features like shape or size to further discriminate between targets and clutter), then the performance of the tracker will be better.

D. Track-Before-Fuse Comparison

The tracker presented in this paper maximizes the likelihood of the target state directly in the global three-dimensional space. An alternative approach is to make a batch estimate of the track in the measurement space of each individual sensor and then fuse them into a batch estimate in the global three-dimensional space. For a notional representation of this process, refer to Fig. 14. If the sensors are cameras, one could obtain batch estimates of the target track in the image space of each camera (red segments in Fig. 14) using measurements from only the camera under consideration. Then, given the poses of a pair of the cameras, one could triangulate to a corresponding batch estimate in the global space (orange segments in Fig. 14). This approach may be con-

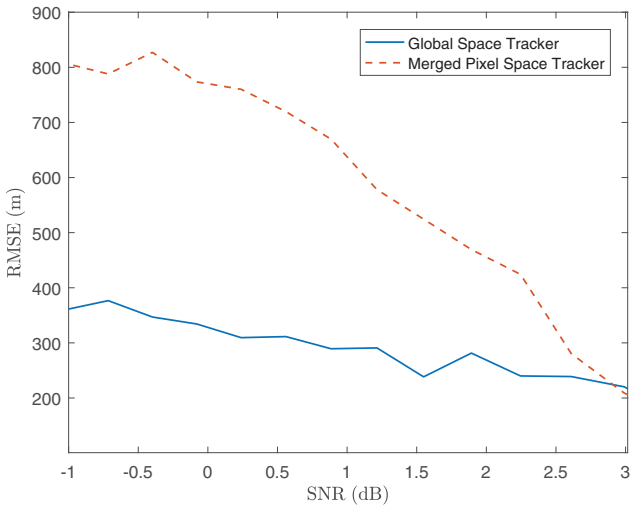


Fig. 15. The RMS position error for the global space tracker (solid line) and the RMSE for the image space tracker (dotted line) plotted for implied measurement extraction SNR values. Errors are averaged over 100 Monte Carlo runs and averaged over time.

sidered desirable in situations where communication of data is restricted: In most realistic settings, the communication cost for sensors to send batches of measurement scans to a central maximizer is higher than if each sensor sends an ML estimate of a track segment. However, this method has the obvious drawback of not having access to the discriminatory power of the "hill finder" algorithm in Section IV.A since the premise is that measurement scans have not been sent to a central location.

We use this decentralized method as a benchmark comparison at various SNR values. The expectation is that the method presented in this paper should be more robust—in terms of mean squared estimation error—as the SNR decreases. This intuitive expectation is due to the fact that both the hill finder and the centralized likelihood evaluation use the relative geometric arrangement of the sensors, which creates a triangulation effect. In terms of the log likelihood, the effect is that the hills climbed by the central tracker are much steeper and taller than the hills in the individual sensor space trackers, which results in a better ML estimate from the central tracker.

Monte Carlo simulations of the scenario described in the previous section are performed as the SNR varies from -1 to 3 dB. As the scenario progresses, both types of tracking are performed simultaneously—tracking according to the algorithm presented in this paper and ML-PMHT tracking in the image spaces of the individual sensors and then fusing into three-dimensional space. The RMSEs obtained with the two different methods are plotted versus the SNR in Fig. 15.

E. JPDA Comparison

A scenario with two targets is simulated in order to test the multi-target capabilities of the ML method presented in this work. For the sake of comparison, the scenario is also subjected to the JPDA tracker, which is a typical recursive maximum *a posteriori* method that has been adapted in [20] to work with multiple sensors.

The sensor arrangement remains the same as the simulation in Section V.A. Along with the original target from the scenario in Section V.A, an additional target is simulated that starts from rest at the point (1000.0, 0.0, 0.0) in the global space and accelerates linearly to arrive at the point $(-1000.0, 0.0, 2000.0)$ at the end of the scenario. The linear path of the second target remains well-resolved from the first target in the three-dimensional global space, although the targets may become unresolved in the two-dimensional measurement space of any one sensor. Although both targets violate the constant velocity assumption of the ML-PMHT, it remains a close approximation over the batch duration, which is shortened to $N^{\mathrm{b}} = 12$ for this simulation. The batch slide is also reduced to 1. That is, a batch estimate is obtained at every time step based on the current scans along with the scans from the past 11 time steps. This ensures that the ML-PMHT is informed by every point obtained by the hill finder routine.

The measurement error is the same as in Section V.A ($\sigma_\theta = \sigma_\phi = 0.25^\circ$). The measurement extraction is assumed to operate such that, for each target, the probability of receiving a hit at each sensor is $p_{\mathrm{d}} = 0.5$, and the expected number of clutter measurements in each scan of each sensor is $\lambda = 50$.

For the JPDA, a single linear white noise acceleration model is assumed for the targets with the process noise parameterization being informed by the maximum acceleration exhibited by the true targets.

Both the JPDA and the ML-PMHT are initialized with a random joint state estimate distributed (with large variance) around the truth, although the ML-PMHT is also allowed immediate access to the hill finder as well, so it has a chance of finding a better initialization for its first batch estimate.

Fig. 16 shows the root-mean-squared position estimate error versus the time step of the scenario averaged over both targets. The benefit of the point finder employed by the ML-PMHT is evident at the beginning of the scenario—while both trackers are given the same random initialization, the ML-PMHT immediately makes use of the point finding, which, on average, results in faster convergence to the "steady-state" performance. A test for statistical significance of the Monte Carlo comparison, as given in [4], is as follows. For a particular Monte Carlo run mc, the RMSE from each of the tracking methods is averaged over time, and the difference between the averages is noted as Δ_{mc}. After all runs are completed, the sample mean and sample standard deviation of the "deltas" are computed, and the significance of the comparison is taken to be the sample mean divided by the sample standard deviation. A value $\gg 2$ is taken to indicate that the performance difference is present in a significant number of runs. The comparison in this simulation was found to have a significance of 4.7.

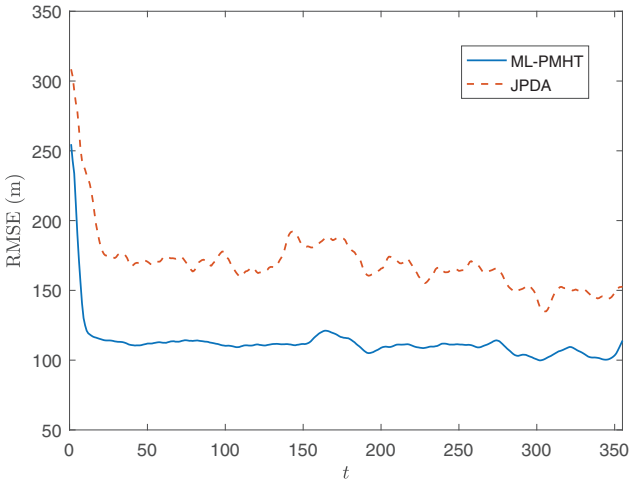


Fig. 16. The RMS position error for the maximum-likelihood tracker (solid line) and the RMSE for the JPDA (dotted line) plotted versus time step. Errors are averaged over 100 Monte Carlo runs and averaged over both targets.

Note that this scenario simulates a relatively low SNR, and previous work has suggested that the ML-PMHT likelihood formulation is more suitable than the JPDA in such settings [14], [21].

VI. CONCLUSION

The algorithm presented in this work is found to outperform a similar track-before-fusing algorithm at reasonably low SNR levels. This result is intuitively predictable given that the global space tracker is taking advantage of knowing the geometric layout of the sensors relative to each other when evaluating the observation likelihood. It is also shown that there is a performance "sweet spot" for the underlying measurement extraction (the process by which "point hits" are declared) where the target detection probability is high enough for the likelihood maximization to be effective, but the level of clutter does not overwhelm the batch estimation.

The "point finding" method presented in Section IV.A, culminating in the list of weighted points in (46), is not fundamentally tied to the ML-PMHT. Instead, it represents a standalone method for obtaining a list of points in three-dimensional Cartesian space by fusing "likely-to-be-related" pairs of two-dimensional line-of-sight measurements from different sensors. The extracted points are used in this work to initialize the likelihood maximization, and are found to help the ML-PMHT stay on track. However, the point extraction could also be used in other fundamentally different settings. For instance, the "target birth" process and two-point-differencing, which is essential to track initialization in the multi-Bernoulli formulation, could be informed by the presented point finding method in a tracking scenario where targets exist in $\mathbb{R}^3$ and multiple sensors report two-dimensional measurements.

The formulation of the batch ML-PMHT presented in this work allows for tracking multiple targets, and is compared to the JPDA for a two-target scenario with high clutter level and low probability of target detection. The ML-PMHT is found to outperform the JPDA in this case, which agrees with results in other works. In terms of average computational cost, the ML-PMHT avoids the expensive data-association step required by the JPDA, but encounters the curse of dimensionality in the parameter space when dealing with multiple targets. Overall, the results suggest that the ML-PMHT is a desirable option in settings where the user wishes to refine track estimates for relatively few targets in the presence of relatively high levels of clutter and low detection probability. Future work will insert the results of the ML-PMHT to a generalized likelihood ratio test for target existence using the threshold-setting techniques in [18], and will compare to an automatically track-managed approach such as the multi-Bernoulli filter [15].

APPENDIX A JACOBIAN COMPUTATION FOR ILS

Each iteration of the ILS estimator requires computation of (41): the Jacobian matrix of the stacked measurement function of sensors a and b with respect to Cartesian position of a target in the global space. In keeping with the notation used in Sections II.B and IV.A.1, this matrix is given by

$$\mathbf{H}_q = \begin{bmatrix} \frac{\partial \bar{\theta}_a}{\partial X} & \frac{\partial \bar{\theta}_a}{\partial Y} & \frac{\partial \bar{\theta}_a}{\partial Z} \\ \frac{\partial \bar{\phi}_a}{\partial X} & \frac{\partial \bar{\phi}_a}{\partial Y} & \frac{\partial \bar{\phi}_a}{\partial Z} \\ \frac{\partial \bar{\theta}_b}{\partial X} & \frac{\partial \bar{\theta}_b}{\partial Y} & \frac{\partial \bar{\theta}_b}{\partial Z} \\ \frac{\partial \bar{\phi}_b}{\partial X} & \frac{\partial \bar{\phi}_b}{\partial Y} & \frac{\partial \bar{\phi}_b}{\partial Z} \end{bmatrix}_{\mathbf{X}=\hat{\mathbf{p}}_q}, \tag{53}$$

where X, Y, Z are the individual Cartesian coordinates of the position vector $\mathbf{X}$ in the global reference frame. Using the formula in (14), which transforms a point in global coordinates into a point in the coordinate system of sensor i, and, specifically, the rotation matrix defined by (15), it can be shown that

$$\mathbf{H}_q = \begin{bmatrix} \mathbf{H}_{q,a}\ \boldsymbol{\Gamma}'_a \\ \mathbf{H}_{q,b}\ \boldsymbol{\Gamma}'_b \end{bmatrix}, \tag{54}$$

where $\boldsymbol{\Gamma}'_i$ is the transpose of the rotation matrix of sensor i, and we have used the property of a rotation matrix that its inverse is equal to its transpose. The other sub-matrix terms are given by

$$\mathbf{H}_{q,i} = \begin{bmatrix} \frac{\partial \bar{\theta}_i}{\partial x_i} & \frac{\partial \bar{\theta}_i}{\partial y_i} & \frac{\partial \bar{\theta}_i}{\partial z_i} \\ \frac{\partial \bar{\phi}_i}{\partial x_i} & \frac{\partial \bar{\phi}_i}{\partial y_i} & \frac{\partial \bar{\phi}_i}{\partial z_i} \end{bmatrix}_{\mathbf{x}_i=\boldsymbol{\Gamma}'_i(\hat{\mathbf{p}}_q-\boldsymbol{\xi}_i)}, \tag{55}$$

which is the Jacobian of the measurement vector of sensor i with respect to a point in the coordinate system of sensor i, evaluated at the current ILS estimate transformed to the coordinate system of sensor i. Taking the corresponding partial derivatives of (9) yields the

individual terms

$$\frac{\partial \bar{\theta}_i}{\partial x_i} = 0, \tag{56}$$

$$\frac{\partial \bar{\theta}_i}{\partial y_i} = \frac{z_i}{y_i^2 + z_i^2}, \tag{57}$$

$$\frac{\partial \bar{\theta}_i}{\partial z_i} = \frac{-y_i}{y_i^2 + z_i^2}, \tag{58}$$

$$\frac{\partial \bar{\phi}_i}{\partial x_i} = \frac{\sqrt{y_i^2 + z_i^2}}{x_i^2 + y_i^2 + z_i^2}, \tag{59}$$

$$\frac{\partial \bar{\phi}_i}{\partial y_i} = \frac{-x_i y_i}{(x_i^2 + y_i^2 + z_i^2)\sqrt{y_i^2 + z_i^2}}, \tag{60}$$

$$\frac{\partial \bar{\phi}_i}{\partial z_i} = \frac{-x_i z_i}{(x_i^2 + y_i^2 + z_i^2)\sqrt{y_i^2 + z_i^2}}, \tag{61}$$

where x_i, y_i, z_i are the individual Cartesian coordinates in the sensor i reference frame. As indicated in (55), for iteration q of the ILS, these terms are evaluated at the coordinates of $\mathbf{x}_i$ given by transforming the current ILS estimate into the reference frame of sensor i with

$$\mathbf{x}_i = \mathbf{\Gamma}_i'(\hat{\mathbf{p}}_q - \boldsymbol{\xi}_i), \tag{62}$$

which, when evaluated for both sensors a and b, will yield $\mathbf{H}_q$ via (54). Note that the notation used here is as in Section IV.A.1 where, since the ILS is being run using a single pair of measurements taken at a single point in time, the time indexing has been omitted. However, in general, the sensor rotation matrices $\mathbf{\Gamma}_i$ and positions $\boldsymbol{\xi}_i$ will be time-dependent.

APPENDIX B ILS INITIALIZATION AND A COMPUTATION-SAVING TRICK

The initial estimate $\hat{\mathbf{p}}_0$ given to the ILS estimator can be evaluated in various ways, and different works differ in the initialization method. In this work, we found that an initialization based on the closest approach between the two line-of-sight measurements worked very well. As discussed in Section IV.A.1, the closest approach will have been previously evaluated for any measurement pair being used in the ILS estimate. Thus, it costs no extra computation to compute the initial estimate as

$$\hat{\mathbf{p}}_0 = \frac{\rho_a}{\rho_a + \rho_b}(\mathbf{p}_b^{\text{md}} - \mathbf{p}_a^{\text{md}}) + \mathbf{p}_a^{\text{md}}, \tag{63}$$

where the positions $\mathbf{p}_i^{\text{md}}$ for $i = a, b$ are the end points of the minimum distance segment connecting the lines-of-sight indicated by measurements a and b evaluated via (30). And, ρ_a, ρ_b are scalars given by (32) and (33), respectively, which provide the Cartesian distance along the lines-of-sight at which the minimum distance occurs. Thus, (63) is a point on the minimum distance segment that lies proportionally closer to the line-of-sight from the sensor that is physically closer to the closest approach. The reasoning is that the farther the sensor is from the target, the farther the corresponding line-of-sight measurement will deviate from the target (in Cartesian distance) due to some fixed measurement error. Thus, when considering a pair of sensors, the line-of-sight from the sensor closer to the target tends to be—in an expected value sense—more "trustworthy" in terms of its Cartesian deviation from the target. This reasoning assumes that the measurement errors in each sensor are identically distributed; if not, then the reasoning becomes less logical. The ILS estimate takes into account non-identical measurement error distributions, however, and (63) still serves as a good initialization in such cases.

If the measurement errors from both sensors *are* identically distributed, the position given by (63) has been found to be a relatively good estimate of the target position, without even running ILS. Of course, ILS will always provide further refinement of the estimate. However, under certain conditions, the composite measurements obtained in Section IV.A.1 via ILS can be replaced by simple evaluation of (63) for any measurement pairs that pass the elimination tests. This would only be done in the name of saving the computation power involved in running ILS for each composite measurement, and if one has computation power to spare, it is not a recommended compromise. However, in simulations with a large number of sensors ($N^{\text{s}} \geq 3$) and in which all pairs of sensors have good cross-range confirmation (see geometric dilution of precision in [4]), composite measurements via (63) were found to serve the purposes of this algorithm rather well. This is due to the fact that the composite measurements, regardless of the level of refinement, are, in turn, used only for the initialization of the batch estimate, which is further refined via the hill climber. Thus, any lack of refinement in the composite measurements from pairs of sensors is quickly made up for once the batch ML estimation is started using all sensors. Qualitatively speaking, this is a good compromise for scenarios with many sensors and small measurement errors, becoming an increasingly poor practice with increasing measurement error and/or decreasing number of sensors.

If the user chooses to make the compromise laid out above, the final thresholding test after ILS is run in Section IV.A.1 can still be performed by replacing $\hat{\mathbf{p}}$ in (35) with $\hat{\mathbf{p}}_0$ from (63). However, no quantitative statements can be made concerning the distribution of the resulting normalized measurement error squared ϵ_0, regardless of whether or not the measurement pair originated from a common target. That is, under the hypothesis that the measurement pair did originate from a common target, it can no longer be claimed that the normalized measurement error squared is Chi-squared distributed. However, referring to Fig. 17, it can be seen that the shapes of the empirically obtained distributions still lend themselves to a simple test by setting a threshold such that, if the normalized estimation error squared

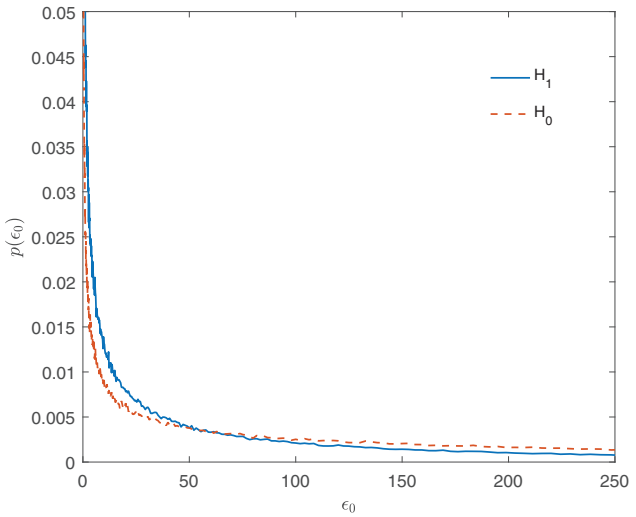


Fig. 17. The empirical distributions of the normalized measurement error squared resulting from the unrefined composite measurement under the binary hypotheses: H_0: the line-of-sight measurements used are unrelated and H_1: the line-of-sight measurements used originated from the same target.

exceeds the threshold, the composite measurement is rejected as being related to a target. However, comparing to the distributions in Fig. 6, it can be seen that the threshold used should be significantly larger than the one used if the composite measurements are refined via ILS. Furthermore, since the distribution under H_1 is no longer Chi-squared, the convenient score based on the Chi-squared cdf in (44) is no longer valid for finalized composite measurements. They must be ranked objectively according to their normalized measurement error squared directly, with smaller errors being considered better.

REFERENCES

[1] S. Avidam and A. Shashua
"Trajectory triangulation: 3D reconstruction of moving points from a monocular image,"
IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 4, pp. 348–357, Apr. 2000.

[2] D. Avitzour
"A maximum likelihood approach to data association,"
IEEE Trans Aerosp. Electron. Syst., vol. AES-28, no. 2, pp. 560–565, Apr. 1992.

[3] B. Balasingam, Y. Bar-Shalom, P. Willett, and K. Pattipati
"Maximum likelihood detection on images," in *Proc. 20th Int. Conf. Inf. Fusion*, Jul. 2017, pp. 1–8.

[4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software. Hoboken, NJ, USA: Wiley, 2001.

[5] H. Coxeter
Projective Geometry. New York, NY, USA: Springer, 2003.

[6] A. Garcia-Fernandez, L. Svensson, J. Williams, Y. Xia, and K. Granstrom
"Trajectory Poisson multi-Bernoulli filters,"
IEEE Trans. Signal Process., vol. 68, pp. 4933–4945, Aug. 2020.

[7] E. Giannopoulos, R. Streit, and P. Swaszek
"Probabilistic multi-hypothesis tracking in a multi-sensor, multi-target environment,"
in *Proc. 1st Australian Data Fusion Symp.*, Nov. 1996, pp. 184–189.

[8] R. Hartley and A. Zisserman
Multiple View Geometry in Computer Vision. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[9] M. Kowalski, P. Willett, T. Fair, and Y. Bar-Shalom
"CRLB for estimating time-varying rotational biases in passive sensors,"
IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 1, pp. 343–355, Feb. 2020.

[10] M. Krieg and D. Gray
"Multi-sensor, probabilistic multi-hypothesis tracking,"
in *Proc. 1st Australian Data Fusion Symp.*, Nov. 1996, pp. 153–158.

[11] Q. Lu, Y. Bar-Shalom, P. Willett, and B. Balasingam
"Measurement extraction for a point target from an optical sensor,"
IEEE Trans Aerospace Electron. Syst., vol. 54, no. 6, pp. 2735–2745, Apr. 2018.

[12] K. G. Murty
"An algorithm for ranking all the assignments in order of increasing cost,"
Operations Res., vol. 16, no. 3, pp. 682–687, May 1968.

[13] R. Osborne and Y. Bar-Shalom
"Statistical efficiency of composite position measurements from passive sensors,"
IEEE Trans. Aerosp. Electron. Syst., vol. 49, no. 4, Oct. 2013, pp. 2799–2806.

[14] C. Rago, P. Willett, and R. Streit
"A comparison of the JPDAF and PMHT tracking algorithms,"
in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, May 1995, pp. 3571–3574.

[15] S. Reuter, B. T. Vo, B. N. Vo, and K. Dietmayer
"The labeled multi-Bernoulli filter,"
IEEE Trans. Signal Process., vol. 62, no. 12, pp. 3246–3260, Jun. 2014.

[16] R. Streit and T. Luginbuhl
"A probabilistic multi-hypothesis tracking algorithm without enumeration and pruning," in *Proc. 6th Joint Service Data Fusion Symp.*, Jun. 1993.

[17] R. Streit and T. Luginbuhl
"Probabilistic multi-hypothesis tracking," Naval Undersea Warfare Center, Newport, RI, USA, Tech. Rep. 10,428, Feb. 1995.

[18] S. Schoenecker, P. Willett, and Y. Bar-Shalom
"Extreme-value analysis for ML-PMHT, Part 1: Threshold determination for false track probability,"
IEEE Trans. Aerosp. Electron. Syst., vol. 50, no. 4, pp. 2500–2514, Oct. 2014.

[19] D. Sunday
Practical Geometry Algorithms With C++ Code.
Independently published, 2021.

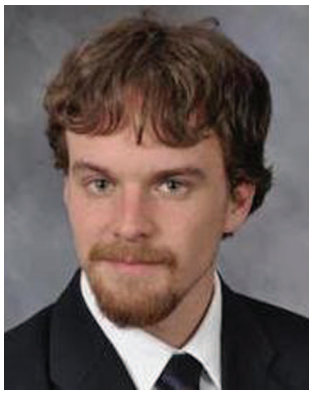
[20] J. K. Tugnait
"Tracking of multiple maneuvering targets in clutter using multiple sensors, IMM, and JPDA coupled filtering,"
IEEE Trans. Aerosp. Electron. Syst., vol. 40, no. 1, Jan. 2004, pp. 320–330.

[21] P. Willett and S. Coraluppi
"MLPDA and ML-PMHT applied to some MSTWG data,"
in *Proc. 9th Int. Conf. Inf. Fusion*, Feb. 2007, pp. 1–8.

[22] Euler angles. (2016). Accessed: Jan. 10, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Euler_angles).

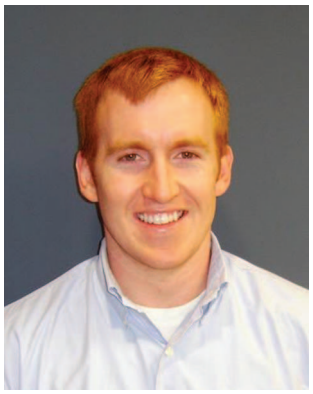
[23] Pinhole camera model. (2018). Accessed: Feb. 10, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Pinhole_camera_model.

[24] scipy.optimize.minimize. (2021). Accessed: Sep. 1, 2021. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html



Zachariah Sutton received the B.Sc. degree in electrical engineering from the University of Connecticut, Storrs, CT, USA, in 2016. He is currently working toward Ph.D. degree in the Electrical and Computer Engineering Department, University of Connecticut. His research interests include signal processing, statistical modeling, and target tracking.

Peter Willett (F'03) has been a faculty member in the Electrical and Computer Engineering Department, University of Connecticut, Storrs, CT, USA, since 1986. Since 1998, he has been a Professor. His primary areas of research have been statistical signal processing, detection, machine learning, communications, data fusion, and tracking. He was the Editor-in-Chief for the IEEE Signal Processing Letters from 2014 to 2016. He was the Editor-in-Chief for the IEEE Transactions on Aerospace and Electronic Systems from 2006 to 2011 and AESS Vice President for Publications from 2012 to 2014. From 1998 to 2005, he was an Associate Editor for three active journals—IEEE Transactions on Aerospace and Electronic Systems (for data fusion and target tracking) and IEEE Transactions on Systems, Man, and Cybernetics, and parts A and B. He is remains an Associate Editor for the IEEE AES Magazine. He is a member of the IEEE AESS Board of Governors and of the IEEE Signal Processing Society's Sensor-Array and Multichannel (SAM) technical committee (and is now Chair).



Tim Fair, Toyon Senior Staff Analyst, Deputy Director Signal Processing, Principle Investigator, received the B.Sc. and M.Sc. degrees in electrical and computer engineering from the University of California, San Diego, CA, USA, with focus on signal and image processing, in 2008 and 2009, respectively. He has worked with SAIC and Johns Hopkins Applied Physics Lab, where he developed and analyzed detection, tracking, association, and discrimination algorithms for next-generation ISR platforms and Missile Defense Agency satellite systems and the Navy Aegis Ballistic Missile Defense program. Since joining Toyon in 2012, he has focused on algorithm development for image and video processing. As part of this work at Toyon, Mr. Fair has developed solutions in the fields of target detection and tracking, trajectory estimation, target pose estimation, and more recently on methods for applying deep learning and artificial intelligence in the military sector.

Yaakov Bar-Shalom (F'84) received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion, Haifa, Israel, in 1963 and 1967, respectively, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, USA, in 1970. He is currently a Board of Trustees Distinguished Professor with the ECE Department and Marianne E. Klewin Professor with the University of Connecticut. His current research interests are in estimation theory, target tracking, and data fusion. He has published more than 650 papers and book chapters. He coauthored/edited eight books, including *Tracking and Data Fusion* (YBS Publishing, 2011). He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He served as an Associate Editor for the IEEE Transactions on Automatic Control and Automatica. He was General Chairman of the 1985 ACC, General Chairman of FUSION 2000, President of ISIF in 2000 and 2002, and Vice President for Publications from 2004 to 2013. Since 1995, he has been a Distinguished Lecturer of the IEEE AESS. He is a corecipient of the M. Barry Carlton Award for the best paper in the IEEE TAESystems in 1995 and 2000. In 2002, he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. In 2008, he was awarded the IEEE Dennis J. Picard Medal for Radar Technologies and Applications, and in 2012, the Connecticut Medal of Technology. He has been listed by academic.research.microsoft (top authors in engineering) as #1 among the researchers in aerospace engineering based on the citations of his work. He is the recipient of the 2015 ISIF Award for a Lifetime of Excellence in Information Fusion. This award has been renamed in 2016 as the Yaakov Bar-Shalom Award for a Lifetime of Excellence in Information Fusion. He has the following Wikipedia page: https://en.wikipedia.org/wiki/Yaakov Bar-Shalom.

Fusion of Multipath Data From a Remote Sensor for Object Extraction

ANDREW FINELLI
YAAKOV BAR-SHALOM
PETER WILLETT
FRANCESCO A. N. PALMIERI
BRAHAM HIMED

This paper develops an Object Extraction (OE) algorithm from a remote sensor in the presence of multipath propagation between the sensor and the object. The OE is carried out by estimating the object's motion parameter by fusing the multipath measurements. The signals from the object are assumed to have a low signal-to-noise ratio i.e., the OE has to be done in the presence of numerous spurious detections. This paper also discusses a method to reduce the size of the motion parameter space by considering the object's motion in a non-inertial frame. The object is observed using a measurement model that produces range, azimuth, and range-rate using a multipath refraction model for the signal propagation through the medium. The OE accounts for the multipath environment as the model allows for multiple returns from a single object. Finally, the paper shows the effectiveness of the OE by evaluating the accuracy of the estimation with Monte Carlo simulation.

Manuscript received March 30, 2021; revised October 13, 2021; released for publication December 1, 2021.

A. Finelli, Y. Bar-Shalom, P. Willett, and F. A. N. Palmieri are with the Department of Electrical and Computer Engineering, University of Connecticut, Sorrs, CT 06269, USA (E-mail: andrew.finelli@Uconn.edu, yaakov.bar-shalom@Uconn.edu, Peter.Willett@Uconn.edu, francesco.palmieri@unicampania.it).
B. Himed is with the Distributed RF Sensing Branch, Air Force Research Labratory, Dayton, OH (E-mail: braham.himed@us.af.mil).

I. INTRODUCTION

There have been many approaches to extracting object state or motion parameters from sensor data in a cluttered environment. Some of these solutions use a probabilistic data association filter (PDAF) [7], a multipath PDAF [20], or extensions of the multiple hypothesis tracker (MHT) such as the multiple detection MHT [23]. Our solution is to use the Object Extractor (OE), based on the maximum likelihood probabilistic multiple hypothesis tracker (ML-PMHT). This method creates a likelihood function (LF) for the motion parameters, based on the data and the object's measurement model. This LF is then maximized to produce the most likely set of parameters for that object's motion. This method is attractive because the data association in this algorithm is implicit. Furthermore, as the number of scans is increased, the ML-PMHT is the only algorithm that does not suffer a combinatorial explosion in computational complexity. Lastly, the OE is shown to have the ability to extract very low signal-to-noise ratio (SNR) objects from cluttered observations. This was shown explicitly for multiple objects in a multipath environment with the development and evaluation of the joint multipath ML-PMHT [24].

The present work extends the previous multipath works [8], [21] in two vital ways. The first is that the multipath measurement fusion has been enhanced for realism through the use of coordinate system transformations. This has been exemplified through the use of an atmospheric refraction model [4]. The second notable extension relates to the number of target parameters. Whereas previous treatments explored low-dimensional parameter spaces (straight-line motion), we add acceleration in three dimensions. In our examples, we explore a model with significant centripetal acceleration, drag, and gravity. The drag on these objects will be modeled to increase linearly with its centripetal acceleration, and gravity is constant in magnitude and radially toward the origin. The motion of such objects cannot be approximated as straight line segments and coordinated turns as has often been done in previous works [21] and thus requires a large number of parameters to accurately predict their motion [26]. This has the potential to cause the optimization involved in the OE to become computationally intractable due to ill-conditioning.

The measurement model considered in this work exploits the phenomenon where alterations to the signal return path permits the sensor to detect objects that are beyond the sensor's line of sight. This phenomenon results in the signal propagating along several paths from the originating point to an object and vice versa for the reflected signal. This allows for multiple returns from a single object during one scan, causing an ambiguity as to which detection belongs to which path [27]. Further increasing the difficulty of determining the object's motion parameters, there are false detections causing spurious measurements (clutter) from all paths. The OE

algorithm has been extended to operate in a multiple path formulation, presented in [21]. This extension relies on the fact that the OE natively assumes that any number of measurements may associate to an object and modifies the fusion of measurements by allowing for different measurement models (one for each path) with associated prior probabilities. Another algorithm that takes advantage of the multipath measurements considered in this work is the multiple detection maximum likelihood probabilistic data association (MD-ML-PDA), an extension of the standard maximum likelihood probabilistic data association to associate multiple measurements to a single object. The development and evaluation of the MD-ML-PDA can be found in [25].

This work seeks to solve the problem of motion parameter extraction for an object executing a complex coordinated turn outside line of sight of the sensor and in a multipath environment. In Section II, we describe the motion model which will be considered deterministic and nonlinear during a batch of sensor data. We also define a reference frame conversion that defines the object's motion and allows for the use of a smaller object motion parameter vector during extraction. In Section III, we introduce the multipath measurement model that uses a refraction function to calculate the range and range-rate measurements. We also provide a method for thresholding measurements to limit the computational load of the OE in this section. Section IV describes the operation of the OE algorithm used to process these measurements, and explains how it is extended to the multipath scenario. Finally, in Section V, we present and discuss simulation results to test the algorithm's effectiveness in tracking these objects.

II. OBJECT MOTION MODEL

The kinematic motion model for the objects we consider here is three-dimensional (3D) and uses a modified coordinated turn model. The objects are assumed incapable of thrust during the period of observation and therefore we will not model this. The object is under the influence of gravity that is taken to be constant and equal to that at Earth's surface (i.e., $g = 9.81\ \mathrm{m/s^2}$). We use the Earth-centered, Earth-fixed (ECEF) coordinate system, which is a Cartesian system with the origin at the center of the earth [15]. The motion model must describe the propagation of the position and velocity of the object. We define a stacked vector of these values as

$$\theta(t) = \begin{bmatrix} x(t) & y(t) & z(t) & v_x(t) & v_y(t) & v_z(t) \end{bmatrix}', \quad (1)$$

with the initial values denoted θ_0. The components of an object's acceleration due to radial gravity are

$$g_x(t) = \frac{-gx(t)}{r(t)}, \qquad g_y(t) = \frac{-gy(t)}{r(t)}, \qquad g_z(t) = \frac{-gz(t)}{r(t)}, \quad (2)$$

where $r(t)$ is the distance from the object to the origin, i.e.,

$$r(t) = \sqrt{x^2(t) + y^2(t) + z^2(t)}. \quad (3)$$

Next, the drag component of the object's acceleration has magnitude

$$|a_{\mathrm{drag}(t)}| = \frac{C_d \rho(r) A_X S^2(t)}{2m_t}, \quad (4)$$

where C_d is the drag coefficient, A_X is the object's cross-sectional area, m_t is the object's mass, and S is the speed of the object, namely,

$$S(t) = \sqrt{v_x^2(t) + v_y^2(t) + v_z^2(t)}. \quad (5)$$

In an effort to limit the computational complexity, we lower the parameter space by assuming parameters related to drag to be known. During a turn, the drag experienced by an object increases due to the aerodynamic effectors used. We model this by modifying C_d to linearly increase with the centripetal acceleration (other models can be used as well). Specifically, the drag coefficient increases such that an acceleration of $a_c = 10g$ causes the drag to increase by 20%, i.e.,

$$C_d = C_{d0}\left(1 + \frac{a_c}{50g}\right), \quad (6)$$

where C_{d0} is the drag coefficient for the object flying in a straight line. Lastly, $\rho(r)$ is the density of the medium as a function of object's position. The density of the propagation medium is modeled as a first-order differential equation with exponential solution

$$\rho(r) = \rho_0 \exp\left[-\frac{r(t) - R}{r_0}\right], \quad (7)$$

where ρ_0 and r_0 are constants defined for the medium and R is Earth's radius (about 6371 km). With this definition, the acceleration of the object due to drag (in the x-coordinate, but it is similar in y and z-coordinates) is

$$a_{x,\mathrm{drag}} = \frac{-|a_{\mathrm{drag}}(t)| v_x(t)}{S(t)}. \quad (8)$$

Finally, we need to model the centripetal acceleration due to a turn in 3D space. The turn is defined by a vector of orthogonal turn-rates described within the global (ECEF) reference frame and we denote this vector (with units in rad/s) as

$$\Omega^G(t) = \begin{bmatrix} \omega_x^G(t) & \omega_y^G(t) & \omega_z^G(t) \end{bmatrix}', \quad (9)$$

where the superscript G indicates the global reference frame. The way that these turn rates relate to the constant turn rate within the object's reference frame will be described in the next subsection. The centripetal acceleration is then

$$a_c(t) = \Omega^G(t) \times \begin{bmatrix} v_x(t) & v_y(t) & v_z(t) \end{bmatrix}', \quad (10)$$

where $\times$ indicates the vector cross product. This cross product can be expressed using a skew-symmetric matrix

to pre-multiply the velocity vector. Here, this matrix is denoted as $K_\Omega(t)$ and allows the centripetal acceleration to be written as

$$a_c(t) = \underbrace{\begin{bmatrix} 0 & -\omega_z^G(t) & \omega_y^G(t) \\ \omega_z^G(t) & 0 & -\omega_x^G(t) \\ -\omega_y^G(t) & \omega_x^G(t) & 0 \end{bmatrix}}_{K_\Omega(t)} \begin{bmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \end{bmatrix}. \tag{11}$$

The state vector $\theta(t)$ changes according to these time-varying accelerations and can be concisely expressed using a block partitioned matrix as

$$\dot{\theta}(t) = \begin{bmatrix} 0_{3\times3} & 1_{3\times3} \\ \frac{-g}{r(t)} I_{3\times3} & K_\Omega(t) - \frac{-|a_{\text{drag}}(t)|}{S(t)} I_{3\times3} \end{bmatrix} \theta(t). \tag{12}$$

This equation is a non-linear differential equation, so to produce trajectories, we use a fourth-order Runge–Kutta method [22] with a sufficiently small step size δ to approximate the true motion observed during the sampling period of T, i.e.,

$$\theta_T(i) = \theta(iT). \tag{13}$$

Further accelerations, caused by the use of a non-inertial (ECEF) reference frame include the Coriolis effect and a centrifugal acceleration. These accelerations were found to be insignificant compared to the centripetal acceleration from turns (and the other forces considered). They are thereby omitted for the sake of model simplicity, but the correction may be included in future works.

A. Reference Frame Conversions

While the turn rate observed in the global ECEF reference frame is constant in magnitude (i.e., the centripetal acceleration is constant), the component vectors ($\omega_x^G(t)$, etc.) are constantly changing. Furthermore, the object can only perform pitch and yaw turns, and zero roll maneuvers (according to our assumption of a point object). Pitch and yaw turns are represented by turn rates $\omega_x^{\mathcal{O}}$ and $\omega_y^{\mathcal{O}}$, respectively, where the superscript $\mathcal{O}$ indicates the object's reference frame. We wish to stress that these values are assumed to be constant parameters in the object's reference frame during the (relatively short) batch length, so there is no time dependence on these values (compared to the global reference frame values). These can be arranged into a vector, similar to $\Omega^G(t)$, as

$$\Omega^{\mathcal{O}}(t) = \begin{bmatrix} \omega_x^{\mathcal{O}} & \omega_y^{\mathcal{O}} & 0 \end{bmatrix}'. \tag{14}$$

In order to convert between the two reference frames, we use a time-varying conversion matrix. In keeping with convention, as well as our above definitions, the z-axis in the object's reference frame is the direction of its velocity at any time. Formally, the z-axis unit vector in ECEF coordinates is the unit vector of the velocity, i.e.,

$$\vec{1}_{z^{\mathcal{O}}(t)} = \frac{1}{S(t)} \begin{bmatrix} v_x(t) & v_y(t) & v_z(t) \end{bmatrix}' = \vec{1}_{v(t).} \tag{15}$$

We also define the x-axis in the object's reference frame to be the direction of a perpendicular "right turn" vector (in the local vertical plane) from the object's perspective. Specifically, we construct the object's reference frame x-axis in ECEF coordinates using the object's's current velocity and position as follows:

$$\vec{1}_{r(t)} = \frac{1}{r(t)} \begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix}', \tag{16}$$

$$\vec{1}_{x^{\mathcal{O}}(t)} = \frac{\vec{1}_{v(t)} \times \vec{1}_{r(t)}}{\left| \vec{1}_{v(t)} \times \vec{1}_{r(t)} \right|}, \tag{17}$$

where $\times$ denotes the cross-product.

Naturally, the object reference frame y-axis in ECEF coordinates is then calculated as

$$\vec{1}_{y^{\mathcal{O}}(t)} = \vec{1}_{z^{\mathcal{O}}(t)} \times \vec{1}_{x^{\mathcal{O}}(t)}. \tag{18}$$

It is important to note that the basis vectors in the object's reference frame ($\vec{1}_{x^{\mathcal{O}}(t)}$, $\vec{1}_{y^{\mathcal{O}}(t)}$, $\vec{1}_{z^{\mathcal{O}}(t)}$) are time-varying.

With these axes defined in ECEF coordinates, we can create the matrix that will transform coordinates from the global ECEF reference frame, to the object's reference frame as a matrix containing these rotated basis vectors.

$$R_{G\to\mathcal{O}}(t) = \begin{bmatrix} \vec{1}_{x^{\mathcal{O}}(t)} & \vec{1}_{y^{\mathcal{O}}(t)} & \vec{1}_{z^{\mathcal{O}}(t)} \end{bmatrix}. \tag{19}$$

Finally, we can define the global turn rates in terms of the constant object's reference frame turn rates as

$$\Omega^G(t) = R_{G\to\mathcal{O}}(t)' \Omega^{\mathcal{O}}, \tag{20}$$

where we make use of the fact that $R_{G\to\mathcal{O}}(t)$ is unitary to perform the inverse reference frame change operation.

The above transformation has two useful aspects. The first is that it allows us to describe the motion that is fundamentally a time-varying turn rate vector in the global reference frame as a constant turn rate vector in the object's reference frame and a time-varying, but simple to calculate, transformation matrix. Secondly, for an object where the roll component of the turn rate vector is always zero, this also allows one to circumvent the need to determine a time-varying, 3D global turn rate vector at each scan and instead look to determine the constant two-dimensional turn rate vector in the object's frame of reference. This second aspect leads to a reduction in the number of parameters needed to describe the object's motion in a global reference frame via the addition of a transformation that depends on the motion parameters at each scan, i.e., the object's global velocity and position relate the three non-zero global turn rates with the object's two non-zero local turn rates.

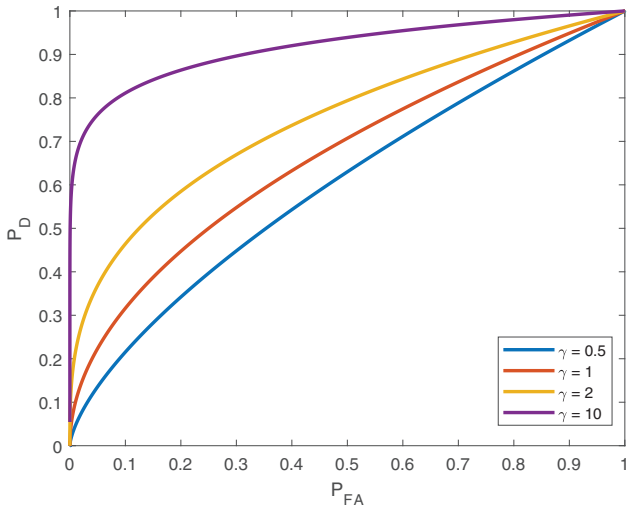


Fig. 1. ROC for the amplitude thresholding described in Section III. The SNRs provided are values of γ as the amplitude ratio. The corresponding decibel values of γ are -3 dB, 0 dB, 3 dB, and 10 dB.

III. SENSOR MEASUREMENT MODEL

Our sensor measurement model will only consider returns from two elevations that produce three paths in a multipath scenario. This is the smallest number of paths for a valid multipath development, but the extension to a greater number of paths is conceptually straightforward and increases the algorithm's complexity linearly. The measurements we receive from the sensor are range (r), range-rate ($\dot{r}$), azimuth (α), and amplitude (A). At time step k, we index the m^{th} set (vector) of measurements as

$$z_m(k) = [r_m(k) \;\; \dot{r}_m(k) \;\; \alpha_m(k) \;\; A_m(k)]'. \tag{21}$$

The amplitudes are generated according to a Swerling I model [3] where the probability density function (pdf) of the amplitude for object absent and object present (in a resolution cell) are, respectively,

$$\mathrm{p}_0(A) = A \exp\left\{-\frac{A^2}{2}\right\} \qquad A \geq 0, \tag{22}$$

$$\mathrm{p}_1(A) = \frac{A}{1+\gamma} \exp\left\{-\frac{A^2}{2(1+\gamma)}\right\} \qquad A \geq 0, \tag{23}$$

where γ is the expected SNR (power ratio) in a resolution cell. We use the amplitude feature to reduce the number of measurements sent to the OE via thresholding. Specifically, for any threshold τ, the probabilities of detection (P_{D}) and false alarm (P_{FA}) are calculated as

$$\mathrm{P}_{\mathrm{FA}} = \int_{\tau}^{\infty} p_0(A)\, dA, \tag{24}$$

$$\mathrm{P}_{\mathrm{D}} = \int_{\tau}^{\infty} p_1(A)\, dA. \tag{25}$$

The detection performance in this situation is easily characterized by the receiver operating characteristic (ROC) curves in Fig. 1. These values are then used to calculate the amplitude pdfs (after thresholding), which are

$$\mathrm{p}_0^{\tau}(A) = \frac{1}{P_{\mathrm{FA}}} p_0(A) \qquad A \geq \tau, \tag{26}$$

$$\mathrm{p}_1^{\tau}(A) = \frac{1}{P_{\mathrm{D}}} p_1(A) \qquad A \geq \tau. \tag{27}$$

The range and range rate portions of each measurement are calculated via a 3D path calculation function. These calculations are performed by an exogenous software with knowledge of the propagation medium [1], [4], [13]. We choose an operating frequency for the sensor of 15 MHz and restrict the number of round-trip paths the signal may travel on to four. White Gaussian noise is added to the range, range rate, and azimuth parts of the measurement vector. Due to the difference between how noise affects different parts of the measurements, we present a modified measurement vector, with the amplitude component removed, to be used separately. Therefore, define the truncated measurement vector containing the kinematic components as

$$z_m^*(k) = [r_m(k) \;\; \dot{r}_m(k) \;\; \alpha_m(k)]'. \tag{28}$$

The covariance matrix of the truncated measurement vector $z_m^*(i)$ is

$$R = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_{\dot{r}}^2 & 0 \\ 0 & 0 & \sigma_\alpha^2 \end{bmatrix}. \tag{29}$$

The noise is additive in the measurement (range, rate-rate, and azimuth) space, and the motion is deterministic given the initial state, therefore the equation for a object-originated measurement is

$$z_m^*(k) = h_\ell(f(\mathbf{x}, k)) + \mathcal{N}([0 \;\; 0 \;\; 0]', R), \tag{30}$$

where $\mathcal{N}(\mu, R)$ is a multi-variate Gaussian with mean μ and covariance matrix R, $h_\ell(\cdot)$ is the measurement function for path l that produced measurement m (supplied by the 3D path calculation function described above, IONORT), applied to the putative location vector produced by the function $f(\mathbf{x}, k)$ (produced by the fourth-order Runge–Kutta method in (13)), and $\mathbf{x}$ is the stacked vector consisting of θ_0 and the non-zero parts of $\Omega^{\mathcal{O}}$, namely,

$$\mathbf{x} = [\theta_0' \;\; \Omega^{\mathcal{O}\prime}]'. \tag{31}$$

IV. OBJECT EXTRACTOR

The OE, an extension of the maximum-likelihood probabilistic multi-hypothesis tracker (ML-PMHT), is a batch estimation algorithm that effectively creates a parameter optimization problem for the OE problem. The OE algorithm constructs the object state LF based on a batch of measurements using a number of assumptions about the data. The maximum of the LF occurs at the

OE estimate of the parameter vector. The assumptions are:

- The number of objects is known (in this work, we consider only one object).
- Any number of measurements may be associated to an object (here, up to four are truly possible).
- The object's motion is deterministic during the (typically short) batch of measurements, given the object's motion parameters.
- False detections are distributed uniformly in space and Poisson in cardinality, i.e., a spatial Poisson process (this may be relaxed).
- Measurement noises are Gaussian, temporally white (conditioned on the parameter vector), and Rayleigh in amplitude (again, the amplitude assumption may be relaxed).

With these assumptions, we build the log-likelihood function (LLF) that can then be maximized over the (initial) object state $\mathbf{x}$ from (31) to provide an estimate $\hat{\mathbf{x}}$ that can be used to determine the object's trajectory (because of the deterministic trajectory assumption).

We will now describe the construction of the LF that forms the basis for the multi-path OE algorithm. First, we define the batch of measurements considered by the OE. Only measurements above the threshold defined above during a window of N_{w} time steps are used. The set of these measurements is defined as

$$\mathbf{Z} \triangleq \left\{ \left\{ z_m^*(i) \right\}_{m=1}^{N_i} \right\}_{i=1}^{N_W}. \tag{32}$$

The LLF of the object motion parameter vector $\mathbf{x}$ based on $\mathbf{Z}$ is then the log of the conditional pdf of the batch. Namely,

$$\begin{aligned} \lambda(\mathbf{x}; \mathbf{Z}) &= \ln\left(\mathrm{p}\left(\mathbf{Z}|\mathbf{x}\right)\right) \\ &= \ln\left\{ \prod_{i=1}^{N_w} \prod_{m=1}^{N_i} \mathrm{p}\left(z_m^*(i)|\mathbf{x}\right) \right\} \\ &= \sum_{i=1}^{N_w} \sum_{m=1}^{N_m} \ln\left\{ \mathrm{p}\left(z_m^*(i)|\mathbf{x}\right) \right\}. \end{aligned} \tag{33}$$

This is a sum over the log of the likelihoods of $\mathbf{x}$ based on each measurement. The LLF is then further expanded by considering each measurement to be from the object-present scenario (hypothesis H_1), with prior probability Π_1, or the object-absent scenario (hypothesis H_0), with prior probability Π_0, independently of all other measurements. The determination of Π_0 and Π_1 is based on the probability of detection within a resolution cell approximately given as

$$\Pi_0 \approx \frac{N_{\mathrm{cells}} \mathrm{P_{FA}}}{N_{\mathrm{cells}} \mathrm{P_{FA}} + \mathrm{P_D}}, \qquad \Pi_1 = 1 - \Pi_0. \tag{34}$$

The LLF can now be written as

$$\begin{aligned} \lambda(\mathbf{x}; \mathbf{Z}) = \sum_{i=1}^{N_w} \sum_{m=1}^{N_m} \ln\big\{ &\Pi_0 \mathrm{p}\left(z_m^*(i)|\mathbf{x}, H_0\right) \\ &+ \Pi_1 \mathrm{p}\left(z_m^*(i)|\mathbf{x}, H_1\right) \big\}. \end{aligned} \tag{35}$$

Disturbance (clutter plus noise) is assumed to be uniformly distributed within the observation volume of size V, i.e.,

$$\mathrm{p}\left(z_m^*(i)|\mathbf{x}, H_0\right) = \frac{1}{V}. \tag{36}$$

Object-originated kinematic measurements are Gaussian with covariance matrix R and the mean of these measurements is the application of the path-dependent measurement function $h_\ell(\cdot)$ to the putative state vector mapped to time i, $f(i, \mathbf{x})$. This allows us to write the conditional pdf of a object based measurement for a specific path ℓ as

$$\mathrm{p}\left(z_m^*(i)|\mathbf{x}, H_1, \ell\right) = \mathcal{N}\left[z_m^*(i); h_\ell\left[f(i, \mathbf{x})\right], R\right], \tag{37}$$

where $\mathcal{N}[z; \mu, R]$ is the Gaussian distribution with variable z, mean μ, and covariance matrix R. Summing over all path likelihoods and multiplying by the path priors gives us the desired part of the likelihood as

$$\mathrm{p}\left(z_m^*(i)|\mathbf{x}, H_1\right) = \sum_{\ell=1}^{N_\ell} \mathrm{p}\left(z_m^*(i)|\mathbf{x}, H_1, \ell\right) \mathrm{p}(\ell) \tag{38}$$

where $\mathrm{p}(\ell)$ is the prior probability of path $\ell \in \{1, 2, \ldots, N_\ell\}$, which we take to be uniform. We also add the amplitude pdfs, which are conditional after thresholding, $\mathrm{p}_0\left(A_m(k)\right)$ and $\mathrm{p}_1\left(A_m(k)\right)$. This is done by assuming the amplitudes to be random and independent of the rest of the measurements, then simply multiplying, i.e.,

$$\begin{aligned} \lambda(\mathbf{x}; \mathbf{Z}) = \sum_{k=1}^{N_w} \sum_{m=1}^{N_m} \ln\Big\{ &\Pi_0 \mathrm{p}_0^\tau\left(A_m(k)\right) \mathrm{p}\left(z_m^*(i)|\mathbf{x}, H_0\right) \\ &+ \Pi_1 \mathrm{p}_1^\tau\left(A_m(k)\right) \mathrm{p}\left(z_m^*(i)|\mathbf{x}, H_1\right) \Big\}. \end{aligned} \tag{39}$$

With the likelihoods defined above, the OE algorithm provides an estimate according to the maximization

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \ \lambda(\mathbf{x}; \mathbf{Z}). \tag{40}$$

It must be stressed that the OE is based on the probabilistic multiple hypothesis tracker (PMHT), and that the PMHT makes the very strong assumption that the provenance (association) of each measurement is independent of that of all others. In the case of standard "hit-based" tracking, this means that zero, or one, or two—or, indeed, all—measurements can associate to the object. In the case of our measurement model, this is extended to allow for multiple measurements to have provenance of (say) upper-path outbound and lower-path return to have non-zero a-priori probability. This

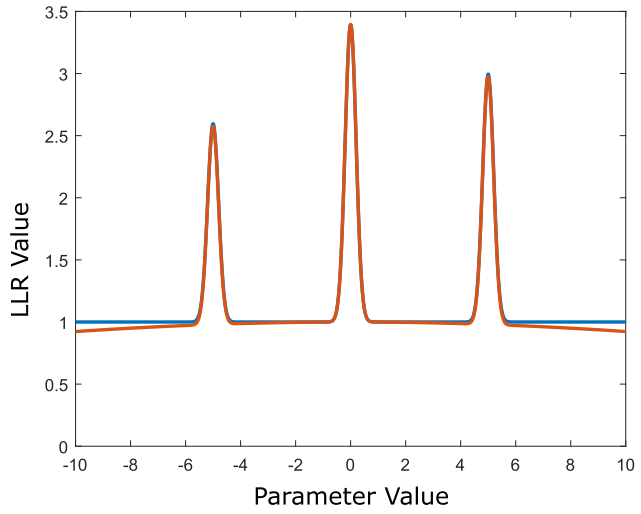


Fig. 2. A notional example of modifying a function that is essentially flat between peaks (in blue) to one with a slight slope introduced by a large, central Gaussian (in orange). We see that the position of the global maximum is unchanged, but any method where candidate points are on the flat surface part (now nearly flat) of the domain will not stall.

formulation leads to computational amenability that the OE exhibits; a multi-scan likelihood evaluation that excludes physically-duplicative events has been shown to be quickly prohibitive due to the combinatorial complexity involved in also considering multiple paths. Fortunately, we find that the "impossible" events are generally afforded small likelihoods, meaning that much practicality is gained with the multipath OE approach.

A. Optimization

The above maximization of the LLF (39) is challenging due, in part, to the fact that the LLF is multimodal. Another challenge exists when calculating the value of the LLF (which is done for many trajectory options) is computationally demanding for even modest sized batches. Our solution requires several different optimization methods in tandem to produce consistent, accurate results.

When performing the global optimization "hill-finding" procedure (and only then), we alter the likelihood surface slightly. Specifically, we alter the flat regions of the LLF to add a relatively small slope such that the optimization stopping criteria will not be easily met. This is shown notionally in Fig. 2. We obtained this slight slope by changing the clutter density from uniform to a truncated Gaussian, whose standard deviation in each dimension is much larger than the volume of the observation area (by an order of magnitude). The truncated Gaussian also has a mean that is placed in the middle of the observation space. The placement of the mean of the truncated Gaussian was found to be of no consequence for the estimate (40) (assuming the variance was sufficiently large and the average value was kept the same). The output of this global optimizer is then given to a gradient ascent method that operates on the original LF to produce the final estimate and likelihood of this estimate. This process is done a maximum of three times, or until the likelihood of the estimate passes a threshold.

The global optimization routine that we used is from the NLopt package [11], and is an implementation of a "controlled random search" with a "local mutation" modification [14]. We also provide our global optimization method with an initial parameter vector around which it will search. This is done by approximately inverting the measurement function and can be found in the Appendix.

TABLE I
Simulation parameters for scenario 1 and scenario 2. All location parameters are in ECEF.

Parameter	Value
R	diag([75m 5m/s 1°])
γ (SNR)	10
τ (Threshold)	3.1
Sensor Pos. (ECEF)	[1450; −2727.2; 5572.2] km
Monte Carlo runs	200
A_X	0.48 m^2
m_t	2000 kg
C_{d0}	0.03
ρ_0	1.2250
r_0	8.5×10^3 m
C_i	6×10^{-5}
P_D (in a cell)	0.65
P_{FA} (in a cell)	0.0082
Π_0, Π_1	0.792, 0.208
N_{cells}	1200

V. SIMULATIONS RESULTS

In this section, we present some interesting scenarios for objects, as well as accuracy statistics pertaining to the performance of our multipath OE. We believe it pertinent to mention that the Cramer–Rao lower bound (CRLB) for a OE can be potentially misleading. The assumption that every measurement can possibly be object-generated is not correct and can produce lower bounds that are not comparable to the state of nature. Specifically, recall the earlier discussion about the OE (and PMHT) association assumptions: in the simulation, we do *not* generate the data according to those assumptions, rather we use the true model that excludes unrealistic associations. Therefore, we will not further discuss the CRLB.

The first simulation tests an object that begins moving perpendicular to the line connecting the object to the sensor. The velocity is also tangent to the Earth with a magnitude of 4 km/s. The object will then perform a 20 m/s^2 pull up and left turn in its own reference frame. The object is observed for 15 s at a sample rate of 1 Hz. The sensor operating frequency is 15 MHz. The rest of

TABLE II
Initial motion parameter vector values and RMSE for scenario 1 at the beginning and end of the OE estimated track. All motion and location parameters are in ECEF. The turn rates are given in respect to the object's reference frame (i.e. the z turn rate is zero and not given).

Parameter	Value	Init. RMSE	Final RMSE
x	1450 km	1275 m	1345 m
y	−2727.2 km	678 m	688 m
z	5572.2 km	139 m	148 m
V_x	3532 m/s	30.83 m/s	36.51 m/s
V_y	1878 m/s	20.9 m/s	18.30 m/s
V_z	0 m/s	6.2 m/s	5.97 m/s
Ω_x^T	23.2 mrad/s	0.63 mrad/s	
Ω_y^T	−65.8 mrad/s	0.73 mrad/s	

the parameters for this scenario can be found in Table I and motion parameters are found in Table II.

Fig. 3 shows measurements from a single Monte Carlo run after thresholding. The scenario was run through the OE algorithm described above to estimate the initial object motion parameters and we examine the root mean square error (RMSE) of this estimate as well as the final position and velocity estimates. The estimated final value is found by propagating the estimated initial value through the deterministic motion equation (12). These errors can be found in Table II and a sample trajectory, both truth and estimated, can be found in Fig 4.

The results for this scenario show that the object position can be estimated with about ~ 1.45 km accuracy of the original position and the velocity estimates are accurate within about 38 m/s. The estimate of the turn rates are accurate to $\sim$ 1 mrad/s on average. The final position RMSE are similar to the initial value RMSE. We will not report the final value of the turn rate as it is constant over a single batch. We also found that the optimizer converged to the correct solution over 94% of the Monte Carlo runs.

A second scenario with different object parameters yielded similar results. The other simulation parameters were kept the same as scenario 1. The values of these parameters and the RMSE associated with them can found in Table III. Fig. 5 shows the estimated trajectory for scenario 2 compared to the truth and the Earth's Surface

Finally, we address the assumption that the drag parameters (other than the speed of the object) are known by exploration method by which we would determine the drag coefficient via maximum likelihood methods. Firstly, Table IV shows the LLF value, evaluated at the true object motion parameters, as a function of the drag coefficient for scenario 2. We see that we can maximize over the value of the drag coefficient, perhaps by testing a discrete number of candidate drag values, and use this value as the true drag coefficient. We can also see from Table IV that a using the incorrect drag causes an increase in error (better to underestimate). In the sce-

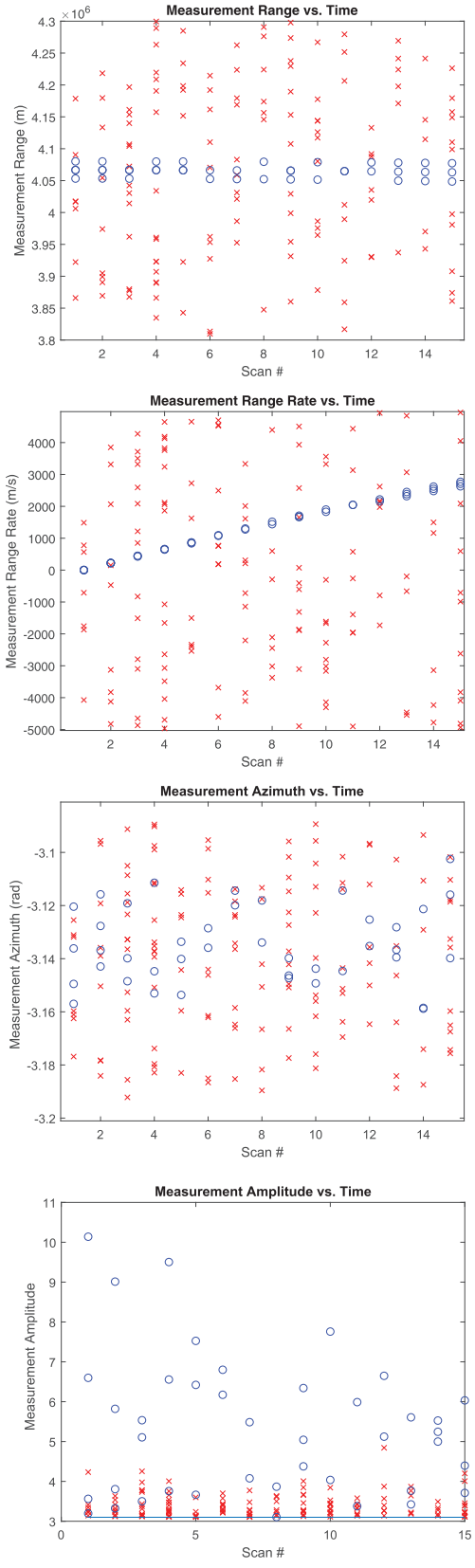


Fig. 3. Sample measurements plotted for scenario 1. The blue markers indicated object originated measurements and red markers indicate disturbance originated (after thresholding). While the originates are shown for illustration only, the OE does not know them. Note that for each object-originated measurement there are about 10 disturbance-originated ones. The OE also does not know the paths corresponding to the object-originated measurements.

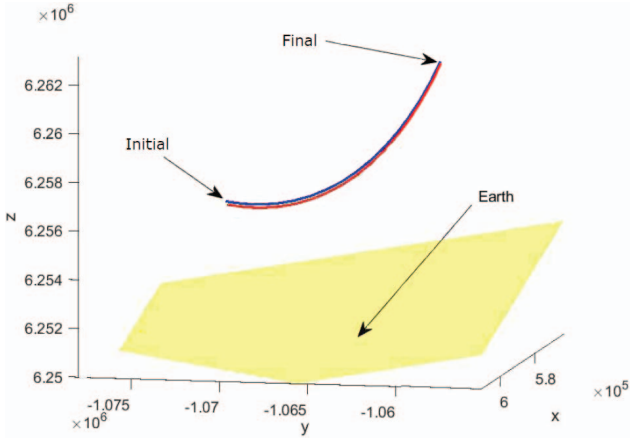


Fig. 4. Sample trajectory for scenario 1 with the true trajectory in blue and the estimated trajectory in red. The yellow plane indicates the Earth surface.

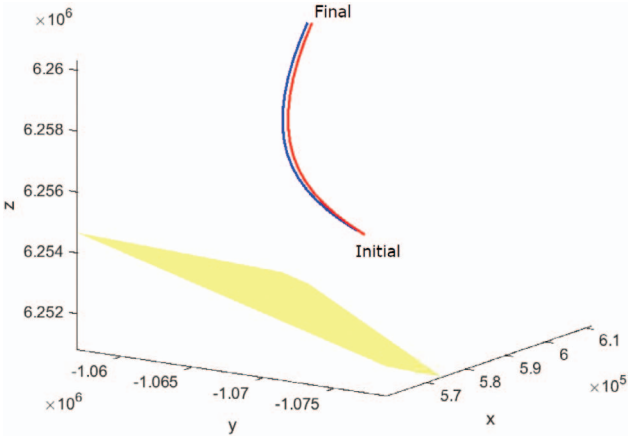


Fig. 5. Sample trajectory for scenario 2 with the true trajectory in blue and the estimated trajectory in red. The yellow plane indicates the Earth surface.

TABLE III
Initial motion parameter vector values and RMSE for scenario 2 at the beginning and end of the OE estimated track. All motion and location parameters are in ECEF. The turn rates are given in respect to the object's reference frame (i.e. the z turn rate is zero and not given).

Parameter	Value	Init. RMSE	Final RMSE
x	571 km	1163 m	1194 m
y	1074 km	616 m	610 m
z	6255 km	142 m	144 m
V_x	3082 m/s	84.5 m/s	97.3 m/s
V_y	2545 m/s	49.3 m/s	40.7 m/s
V_z	155 m/s	5.9 m/s	6.4 m/s
Ω_x^T	−49.4 mrad/s	0.99 mrad/s	
Ω_y^T	−49.4mrad/s	1.2 mrad/s	

nario investigated here (scenario 2), the goal for error in drag estimation should be $\leq 10\%$.

VI. CONCLUSIONS

This work developed a method to track an accelerating object in a multipath environment. We first defined a motion model for objects under the influence of radial gravity and drag that varies with centripetal acceleration. These objects also exhibit rapidly changing motion parameters that preclude the approximation by straight lines even over short batch lengths. We also presented a method by which a conversion between reference frames limits the number of parameters that need to be estimated. We then described the measurement model and thresholding method used to limit the number of measurements delivered to the tracker. The algorithm used is a generalized OE that allows for multipath measurements to be considered and uses a series of optimization algorithms to produce an estimate for the initial object motion parameters during a batch length. Monte Carlo simulations were performed and show the accuracies of position, velocity, and turn rate estimates for the complex scenario we considered.

TABLE IV
A sampling of initial object motion parameter errors when non-exact drag coefficient values are used. We also present the associated log-likelihood values (evaluated at the true object motion parameters) for the mis-matched drag coefficients. We see that larger LLR values correspond to the drag coefficients with less error and a larger error in drag used causes a larger mean squared error in initial object parameter estimation.

% Drag	LLF Val.	$\lvert\mathbf{r_0}\rvert$ Err	$\lvert\mathbf{S_0}\rvert$ Err	$\lvert\mathbf{\Omega}^{\mathcal{O}}\rvert$ Err
70%	448	1963 m	155 m/s	10.0 mrad/s
90%	788	1615 m	66 m/s	3.08 mrad/s
100%	1337	1451 m	38 m/s	0.96 mrad/s
110%	832	1965 m	99 m/s	6.16 mrad/s
130%	510	2537 m	154 m/s	8.32 mrad/s

APPENDIX

A. *Optimizer Initialization by Approximate Inverse Measurement Function*

We developed a method to approximately invert range and azimuth measurements to Earth centered, Earth fixed (ECEF) Cartesian coordinates. The first step in this inversion is to approximate the measurement function, $r_t(\theta_r, \theta_t)$, which calculates the range along a single path from the sensor position (θ_r) to the object's position (θ_t). The approximation used is

$$r_t(\theta_r, \theta_t) \approx C_0 + C_1 D + C_2 a, \tag{41}$$

where D is the great circle distance (GCD) from the sensor to the target, a is the target altitude, and C_n are constants that minimize the error (via a least squares method) in this approximation. There is a different set of coefficients for each possible signal path from the sensor to the object. We have found this approximation to have negligible error over the observation area. The target altitude is unknown, yet necessary for this inversion. In practice, we will test a set of probable altitudes at a fine enough granularity to ensure minimal error. We will

continue this description assuming the altitude being used is the target's true altitude at the measurement time.

The solution involves rotating the global ECEF reference frame such that the sensor lies on the new z-axis (while on the Earth's surface). Solving this equivalent problem is mathematically less complex and the result can be converted back to the standard ECEF reference frame via a rotation matrix. The sensor's position in this new reference frame (θ_r') is

$$\theta_r' = \begin{bmatrix} 0 & 0 & R_E \end{bmatrix}^T, \tag{42}$$

where R_E is the radius of the earth (6371 km). An estimate for the GCD can be found as

$$\hat{D} = \frac{r_t(\theta_r, \theta_t) - C_0 - C_2 a}{C_1}. \tag{43}$$

We note that the noise in the range measurement will affect the accuracy of this estimate. The GCD between the sensor and object is invariant under the change in reference frame, and is calculated as

$$\begin{aligned} D &= R_E \cos^{-1}\left(\frac{\theta_r^T \theta_s}{R_E^2}\right) \\ &= R_E \cos^{-1}\left(\frac{(\theta_r')^T \theta_s'}{R_E^2}\right) = R_E \cos^{-1}\left(\frac{z_s'}{R_E}\right), \end{aligned} \tag{44}$$

where θ_s and θ_s' are the position on the Earth's surface directly below the target in the original and rotated reference frames, respectively. z_s' is the z-coordinate of θ_s'. This shows that the z-coordinate is constant in our solution, i.e., the inverted measurement lies on the plane

$$z_s' = R_E \cos\left(\frac{\hat{D}}{R_E}\right), \tag{45}$$

using the approximate GCD from (43).

The rotation into the new reference frame we use is such that the new x-axis points to the local south in the original ECEF reference frame. Therefore, the azimuth measurement (α) is translated as the clockwise positive angle from the rotated reference frame's negative x-axis (see Fig. 6). Any measurement at this azimuth will lie inside a plane in the rotated reference described as

$$\sin(\alpha) x_s' + \cos(\alpha) y_s' = 0, \tag{46}$$

where x_s' and y_s' are the x-coordinate and y-coordinate of θ_s'.

Using (45) and (46), along with the fact that

$$(x_s')^2 + (y_s')^2 + (z_s')^2 = R_E^2, \tag{47}$$

allows us to solve explicitly for θ_s' as

$$\theta_s' = R_E \left[-\sin\left(\frac{\hat{D}}{R_E}\right)\cos(\alpha) \;\; \sin\left(\frac{\hat{D}}{R_E}\right)\sin(\alpha) \;\; \cos\left(\frac{\hat{D}}{R_E}\right). \right]' \tag{48}$$

This solution is then rotated back into the ECEF reference frame (θ_s) using the sensor spherical position (i.e., $[R_E, \;\; \phi_r, \;\; \psi_r]$) in a rotation matrix as

$$\theta_s = \begin{bmatrix} \cos(\psi_r)\cos(\phi_r) & -\sin(\phi_r) & \sin(\psi_r)\cos(\phi_r) \\ \cos(\psi_r)\sin(\phi_r) & \cos(\phi_r) & \sin(\psi_r)\sin(\phi_r) \\ -\sin(\psi_r) & 0 & \cos(\psi_r) \end{bmatrix} \theta_s'. \tag{49}$$

Finally, the assumed altitude is added onto the surface position to find an approximate inversion for the proposed object position measurement, θ_t.

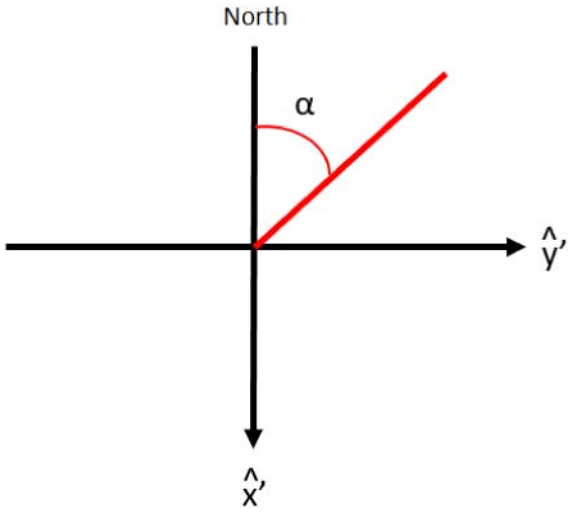


Fig. 6. Representation of the tangent plane to Earth's surface at the sensor in the rotated reference frame. Due North is represented as the negative $\hat{x}'$ direction. The azimuthal angle is given east of due north and is shown in red. The calculation of (49) follows from the diagram.

The estimated locations in (49) use measurements that are corrupted by noise. Specifically, the azimuthal noise causes a multiplicative bias that should and can be corrected for. This is done by multiplying the inverse of the expected value of the bias, as in [2], namely,

$$\theta_s' = R_E \begin{bmatrix} -\sin\left(\frac{\hat{D}}{R_E}\right)\cos(\alpha)\exp\left(\frac{\sigma_\alpha^2}{2}\right) \\ \sin\left(\frac{\hat{D}}{R_E}\right)\sin(\alpha)\exp\left(\frac{\sigma_\alpha^2}{2}\right) \\ \cos\left(\frac{\hat{D}}{R_E}\right) \end{bmatrix}, \tag{50}$$

where σ_α^2 is the azimuthal noise variance (assumed Gaussian). The correction for the range noise is similar, but is not needed for reasonable accuracy.

REFERENCES

[1] A. Azzarone, C. Bianchi, M. Pezzopane, M. Pietrella, C. Scotto, and A. Settimi
"IONORT: A Windows software tool to calculate the HF ray tracing in the ionosphere,"
Comput. Geosci., vol. 42, pp. 57–63, May 2012.

[2] Y. Bar-Shalom, X. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software. Hoboken, NJ, USA: J. Wiley and Sons, 2001.

[3] Y. Bar-Shalom, P. Willett, and X. Tian
Tracking and Data Fusion: A Handbook of Algorithms. YBS Publishing, United States, 2011.

[4] D. Bilitza and B. Reinisch
"IRI 2007: Improvements and new parameters,"
J. Adv. Space Res., vol. 42, no, 4, pp. 599–609, Aug. 2008.

[5] R. Bryan
"Cooperative estimation of targets by multiple aircraft,"
Master's thesis, Air Force Institute of Technology, 1980.

[6] T. Bullock and S. Sangsuk-Iam
"Maneuver detection and tracking with a nonlinear target model,"
in *Proc. 23rd Conf. Decis. Control*, Dec. 1984.

[7] S. Colegrove, S. Davey, and B. Cheung
"PDAF versus PMHT performance on OTHR data,"
in *Proc. Int. Radar Conf.*, Sep. 2003, pp. 560–565.

[8] A. Finelli, Y. Bar-Shalom, P. Willett, F. A. N. Palmieri, and B. Himed
"Target tracking in over the horizon radar,"
in *Proc. SPIE, Signal Process., Sensor/Inf. Fusion, and Target Recognit. XXVIII*, May 2019, doi: 10.1117/12.2519776.

[9] G. Groves, W. Blair, and S. Hoffman
"Some concepts for target trajectory prediction for ship self defense,"
in *Proc. Amer. Control Conf.*, Seattle, WA, Jun. 1995.

[10] J. Hu, X. He, W. Li, H. Ai, H. Li, and J. Xi
"Parameter estimation of maneuvering targets in OTHR based on sparse time-frequency representation,"
J. Syst. Eng. Electron., vol. 27, no. 3, pp. 574–580, Jun. 2016.

[11] S. G. Johnson
"The NLopt nonlinear-optimization package," Available: http://github.com/stevengj/nlopt.

[12] D. Jones, C. Perttunen, and B. Stuckmann
"Lipschitzian optimization without the Lipschitz constant,"
J. Optim. Theory Appl., vol. 79, pp. 157–181, Oct. 1993.

[13] R. M. Jones and J. J. Stephenson
"A versatile three-dimensional computer program for radio waves," U.S. Dept. Commerce, Washington, DC, USA, OT Rep. 75–76, 1975.

[14] P. Kaelo and M. M. Ali
"Some variants of the controlled random search for global optimization,"
J. Optim. Theory Appl., vol. 130, no. 2, pp. 253–264, Jan. 2006.

[15] A. Leick
GPS Satellite Surveying, Hoboken, NJ, USA: J. Wiley and Sons, 2004.

[16] M-Q. Li, J. Xu, X. Zhou, L-C. Qian, T. Long, and M-M. Bian
"OTHR highly maneuvering target detection via generalized randon-fourier transform,"
in *Proc. IEEE Radar Conf.*, 2016.

[17] X. R. Li and V. P. Jilkov
"Survey of fast target tracking part I: Dynamic models,"
IEEE Trans. Aerosp. Electron Syst., vol. 39, no. 4, pp. 1333–1334, Oct. 2003.

[18] J. McCall
"Genetic algorithms for modelling and optimisation,"
J. Comput. Appl. Math., vol. 184, no. 1, pp. 205–222, Dec. 2005.

[19] N. Nabaa and R. Bishop
"Validation and comparison of coordinated turn aircraft turn models,"
IEEE Trans. Aerosp. Electron Syst., vol. 36, no. 1, pp. 250–259, Jan. 2000.

[20] G. Pulford and R. Evans
"A multipath data association tracker for multipath sensorsr,"
IEEE Trans. Aerosp. Electron. Syst., vol. 34, no. 4, pp. 1165–1183, Oct. 1998.

[21] K. Romeo, Y. Bar-Shalom, and P. Willett
"Data fusion with ML-PMHT for very low SNR track detection in an OTHR,"
in *Proc 18th Int Conf Inf Fusion*, Jul. 2015.

[22] C. Runge
"Über die numerische Auflösung von Differentialgleichungen,"
Mathematische Annalen, vol. 46, pp. 167–178, Jun. 1895. doi: 10.1007/BF01446807.

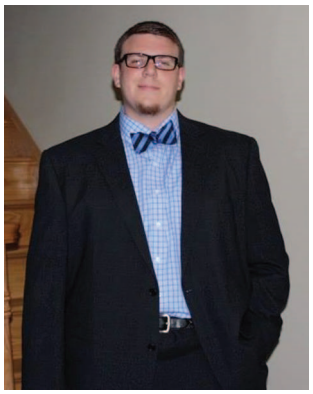
[23] T. Sathyan, T. Chin, S. Arulampalam, and D. Suter
"A multiple hypothesis tracker for multitarget tracking with multiple simultaneous measurements,"
IEEE J. Sel. Topics Signal Process., vol. 7, no. 3, pp. 448–460, Jun. 2013.

[24] X. Tang, Q. Wu, R. Tharmarasa, and T. Kirubarajan
"Multiple detection-aided low-observable track initialization using ML-PDA,"
IEEE Trans. Aerosp. Electron Syst., vol. 53, no. 2, pp. 772–735, Apr. 2017.

[25] X. Tang, R. Tharmarasa, M. McDonald, and T. Kirubarajan
"Multipath maximum likelihood probabilistic multihypothesis tracker for low observable targets,"
IEEE Trans. Aerosp. Electron Syst., vol. 54, no. 1, pp. 502–510, Feb. 2018.

[26] R. Visina, Y. Bar-Shalom, and P. Willett
"Estimation of a random feedback control system with unknown input,"
IEEE Trans. Aerosp. Electron. Syst., Vol. 55, no. 6, pp. 3466–3478, Dec. 2019, doi: 10.1109/TAES.2019.2914539.

[27] J. Wait and A. Murphy
"Multiple reflections between the Earth and the atmosphere in V.L.F. propagation,"
Pure Appl. Geophys., vol. 35, no. 1, pp. 61–72, Feb. 1956.



Andrew Finelli is a Ph.D. student in the Electrical and Computer Engineering Department at the University of Connecticut. He received the B.S. degree in electrical engineering from the University of Connecticut in 2016. His research interests include estimation theory, target tracking, image processing, and machine learning. His doctoral research area is in estimation and signal processing with applications to target tracking.

Yaakov Bar-Shalom received the B.S. and M.S. degrees from the Technion in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in EE. From 1970 to 1976, he was with Systems Control, Inc., Palo Alto, California. Currently he is Board of Trustees Distinguished Professor in the Department of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. His current research interests are in estimation theory, target tracking, and data fusion. He has published over 500 papers and book chapters. He coauthored/edited eight books, including *Tracking and Data Fusion* (YBS Publishing, 2011), He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multi target tracking". He served as Associate Editor of the IEEE Transactions on Automatic Control and Automatica. He was General Chairman of the 1985 ACC. He served as Chairman of the Conference Activities Board of the IEEE CSS and member of its Board of Governors. He served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002, and Vice President for Publications in 2004–2013. In 1987, he received the IEEE CSS Distinguished Member Award. Since 1995, he is a Distinguished Lecturer of the IEEE AESS. He is corecipient of the M. Barry Carlton Award for the best paper in the IEEE TAESystems in 1995 and 2000. In 2002, he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. In 2008, he was awarded the IEEE Dennis J. Picard Medal for Radar Technologies and Applications, and in 2012, the Connecticut Medal of Technology. He has been listed by academic.research.microsoft (top authors in engineering) as #1 among the researchers in Aerospace Engineering based on the citations of his work. He is the recipient of the 2015 ISIF Award for a Lifetime of Excellence in Information Fusion.

Peter Willett received the BASc. degree (engineering science) from the University of Toronto in 1982, and the Ph.D. degree from Princeton University in 1986. He has been a faculty member in the Electrical and Computer Engineering Department at the University of Connecticut since 1986. Since 1998, he has been a Professor, and since 2003, an IEEE Fellow. His primary areas of research have been statistical signal processing, detection, machine learning, communications, data fusion, and tracking. He was the Chief Editor of IEEE AESS Magazine (2018–2021).

Francesco A. N. Palmieri received his Laurea in Ingegneria Elettronica cum laude from Universita' degli Studi di Napoli Federico II, Italy, in 1980. In 1981, he served as a 2nd Lieutenant in the Italian Army in fullfillment of draft duties. In 1982 and 1983, he was with the ITT firms: FACE SUD Selettronica in Salerno (currently Alcatel), Italy, and Bell Telephone Manufacturing Company in Antwerpen, Belgium, as a designer of digital telephone systems. In 1983, he was awarded a Fulbright scholarship to conduct graduate studies at the University of Delaware (USA), where he received the master's degree in applied sciences and the Ph.D. degree in electrical engineering in 1985 and 1987, respectively. He was appointed assistant professor in Electrical and Systems Engineering at the University of Connecticut, Storrs (USA) in 1987, where he was awarded tenure and promotion to Associate Professor in 1993. In the same year, after a national competition, he was awarded the position of Professore Associato at the Dipartimento di Ingegneria Elettronica e delle Telecomunicazioni at Universit degli Studi di Napoli Federico II, Italy, where he has been until October 2000. In February 2000, he was nominated Full Professor of Telecommunication Engineering after a national competition and appointed in November 2000 at Dipartimento di Ingegneria dell'Informazione, Universit della Campania Luigi Vanvitelli, Aversa, Italy. His research interests are in the areas of signal processing, communications, information theory, and machine learning. Francesco Palmieri was the recipient in 1999, with S. Marano and G. Franceschetti, of the S. A. Schelkunof Award for the best paper of the year on IEEE Transactions on Antennas and Propagation. Francesco A. N. Palmieri is also a Visiting Research Scholar at the Electrical and Computer Engineering Department at the University of Connecticut, Storrs CT, USA.

Braham Himed received the engineer degree in electrical engineering from Ecole Nationale Polytechnique of Algiers, Algeria, in 1984, and the M.S. and Ph.D. degrees in electrical engineering from Syracuse University, Syracuse, NY, in 1987 and 1990, respectively. Dr. Himed is a Division Research Fellow with the Air Force Research Laboratory, Sensors Directorate, Multi-Spectral Sensing and Detection Division, Distributed RF Sensing Branch, in Dayton Ohio, where he is involved with several aspects of radar developments. His research interests include detection, estimation, multichannel adaptive signal processing, time series analyses, array processing, adaptive processing, waveform diversity, MIMO radar, passive radar, and over the horizon radar. Dr. Himed is the recipient of the 2001 IEEE region I award for his work on bistatic radar systems, algorithm development, and phenomenology. He is a Fellow of the IEEE and a member of the board of governors of the IEEE Aerospace and Electronics Systems Society (AESS). He is also a past chair of the AESS Radar Systems Panel. He is the recipient of the 2012 IEEE Warren White award for excellence in radar engineering. Dr. Himed is also a Fellow of AFRL (Class of 2013).

Estimation of the Conditional State and Covariance With Taylor Polynomials

SIMONE SERVADIO
RENATO ZANETTI

A novel estimator is presented that expands the typical state and covariance update laws of Kalman filters to polynomial updates in the measurement. The filter employs Taylor series approximations of the nonlinear dynamic and measurement functions. All polynomials (functions approximation, state update, and covariance update) can be selected up to an arbitrary order to trade between filter's accuracy and computational time. The performance of the algorithm is tested in numerical simulations.

Manuscript received January 19, 2021; revised October 24, 2021; released for publication December 1, 2021.

Associate Editor: Paolo Braca.

S. Servadio is with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: simoserv@mit.edu).
R. Zanetti is with the University of Texas at Austin, Austin, TX 78705 USA (e-mail: renato@utexas.edu).

This work was sponsored by the Air Force Office of Scientific Research under Grant FA9550-18-1-0351.

I. INTRODUCTION

Estimation is the process of inferring the value of a quantity of interest from indirect, inaccurate, and noisy observations. When the quantity of interest is the (current) state of a dynamic system, the problem is often referred to as "filtering": The best estimate is obtained by "filtering out" the noise from noisy measurements. The estimate is the output given by an optimal estimator, which is a computational algorithm that processes measurements while maximizing a certain performance index. The optimal estimator makes the best use of the data, of the knowledge of the system, and of the disturbances.

For the well-known linear and Gaussian cases, the posterior distribution remains Gaussian and the Kalman Filter [21], [22] provides the mechanization to calculate its mean and covariance matrix. However, most practical problems are nonlinear in the dynamics and in the measurement equations, leading to non-Gaussian probability density functions (PDFs).

Many techniques have been developed to deal with the nonlinear estimation problem. A simple solution is based on the linearization of the dynamics and measurement equations around the most current estimate. The Extended Kalman Filter (EKF) [13] algorithm applies the KF mechanization to the linearized system. Another well-known technique to account for the system nonlinearities is the unscented transformation. The Unscented Kalman Filter (UKF) [19], [20] is able to better handle the effects of nonlinearities in the dynamics and in the measurements and, typically, achieves higher accuracy and robustness levels when compared to the EKF. The UKF applies the unscented transformation to achieve a more accurate approximation of the predicted mean and covariance matrix. The UKF is a linear estimator, i.e., the estimate is a linear function of the current measurement.

The first-order approximation of the EKF can be extended to higher order Taylor series [10], [13]. Generally, the higher the order of the Taylor series, the better the performance of the filter. The Gaussian Second Order Filter (GSOF) [18] truncates the Taylor series at second order to better account for the system's nonlinearities. Truncating the Taylor series to order c requires knowledge of the estimation error's central moments up to order $2c$ in order to calculate the Kalman gain. For example, the EKF truncates at first order, and it requires knowledge of the covariance matrices. Consequentially, the GSOF requires knowledge of the third and forth central moments of the state distribution. At each iteration, the GSOF approximates the prior PDF as Gaussian so that the third-order central moment is zero and the fourth is easily calculated from the covariance matrix. The GSOF performs a linear update based on a second-order approximation of the posterior estimation error. Linear Gaussian filters exist up to any arbitrary truncation order of the Taylor series approximation of the dynamic/measurement functions [34], [35].

Other linear filters make different types of approximations, such as Gaussian quadrature (QKF) [3], spherical cubature (CKF) [2], ensemble points (EnKF) [42], central differences (CDKF) [32], and finite differences (DDKF) [31].

All of the filters mentioned above are linear estimators, i.e., the estimate is a linear function of the current measurement. The conditional mean, which is the optimal minimum mean square error (MMSE) solution, is typically some unknown nonlinear function of the measurement whose exact computation is usually not feasible. A linear estimator, even when accounting for the nonlinearities of the measurement function, is typically outperformed by nonlinear estimators such as the Gaussian Sum Filter (GSF) [1], [40] or Particle Filters (PFs) such as Bootstrap PF (BPF) [15], Marginalized PF (MPF) [33], Auxiliary PF (APF), Unscented PF (UPF) [44], Gaussian PF (GPF) [16], and Monte Carlo Filter PF (MCFPF) [29].

Ref. [14] derives the evolution of the conditional mean, covariance, and higher order moments of a dynamic system subject to continuous measurements. To make the solution practical, the nonlinear dynamic and measurement equations are approximated with Taylor series expansions.

Another, less studied, approach to nonlinear filtering is to expand the linear update structure to a polynomial update function of the measurement. De Santis *et al.* [11] propose an augmented state to obtain a polynomial update but preserving the linear update structure. Their work augments the measurement vector with its square to form a quadratic update [11] and was extended to polynomial updates [8]. Li *et al.* [23] propose to augment the measurement vector with uncorrelated nonlinear conversions. Similarly to [8] and [11], Liu *et al.* [26] obtain a nonlinear estimator preserving the linear structure of the measurement update. The mean square error (MSE) can be minimized by an optimal selection of the uncorrelated functions [24]. Later, Zhang and Lan [24] merged with the GSF mathematics [46]. Servadio and Zanetti [36] also implemented a quadratic update (extendable to polynomial update of any order) based on Taylor series expansions. The polynomial update requires knowledge of high-order central moments, and [36] carries these moments, exactly like the EKF carries mean and covariance. The computational demand of carrying higher order central moments (propagating forward in time and updating with measurement data) grows quickly with the truncation order of the Taylor series, the size of the state vector, and the order of the polynomial update. Ref. 38 performs a polynomial update without carrying the higher order central moments and, hence, reduces overall computational cost by approximating non-Gaussian distributions as polynomial transformation of Gaussian random variables. In doing so, all high-order central moments are easily and efficiently calculated in a closed form. Consequently, in [38], polynomial updates can be performed much more efficiently than in [36].

The update methodologies presented in [8], [11], [23], [36], and [38] produce a more precise state estimate than those produced by a linear state update. This work introduces a higher order update for the covariance matrix as well as for the state update, which results in a more accurate quantification of the uncertainty associated with the estimate. In turn, the more accurate uncertainty representation produces a more accurate estimator and, hence, a reduced estimation error.

This article is structured in the following way. First, a short background section highlights the novel contributions of the work. This is followed by the development of the new methodology and by applications to three numerical examples. Lastly, conclusions are drawn.

II. BACKGROUND

The linear update rule for mean $\hat{\mathbf{x}}^+$ and covariance matrix $\mathbf{P}^+_{\mathbf{xx}}$ are given by

$$\hat{\mathbf{x}}^+ = \hat{\mathbf{x}}^- + \mathbf{K}(\tilde{\mathbf{y}} - \hat{\mathbf{y}}^-), \tag{1}$$

$$\mathbf{P}^+_{\mathbf{xx}} = \mathbf{P}^-_{\mathbf{xx}} - \mathbf{K}\mathbf{P}_{\mathbf{yy}}\mathbf{K}^T, \tag{2}$$

where $\mathbf{K}$ is the Kalman gain, $\tilde{\mathbf{y}}$ is the measurement outcome, $\hat{\mathbf{y}}^-$ is the predicted measurement mean, $\hat{\mathbf{x}}^-$ is the prior mean, $\mathbf{P}^-_{\mathbf{xx}}$ is the covariance of the state, and $\mathbf{P}_{\mathbf{yy}}$ is the covariance of the measurement. The above equations are optimal in an MMSE only when the prior distribution and the measurement are jointly Gaussian (which implies a linear relation between the two). In general, the MMSE estimate is the conditional mean, an unknown and typically nonlinear function of the measurement outcome; Equation (1) is the statistical linear regression of the conditional mean [25], that is to say, Equation (1) is the best linear fit of the conditional mean with respect to a MSE performance index

$$\hat{\mathbf{x}}^+ \approx \mathbb{E}\left\{\mathbf{x}\middle|\mathbf{y} = \tilde{\mathbf{y}}\right\},$$

where the approximation holds to first order. Equation (2), on the other hand, is the total covariance of the estimation error:

$$\mathbf{P}^+_{\mathbf{xx}} = \mathbb{E}\left\{(\mathbf{x} - \hat{\mathbf{x}}^+)\,(\mathbf{x} - \hat{\mathbf{x}}^+)^T\right\},$$

but it is also the best constant approximation of the conditional covariance of the state given the measurement, also in an MSE sense.

$$\mathbf{P}^+_{\mathbf{xx}} \approx \mathbb{E}\left\{(\mathbf{x} - \mathbb{E}\{\mathbf{x}\})\,(\mathbf{x} - \mathbb{E}\{\mathbf{x}\})^T\middle|\mathbf{y} = \tilde{\mathbf{y}}\right\},$$

where the approximation holds to zeroth order.

For nonlinear dynamics/measurements, the linear update equations above are not fully recursive, and processing nonlinear measurements as a batch is more accurate than processing them individually [36]. For nonlinear systems, Bayes' rule can be applied recursively to

obtain an optimal estimator, that is to say, the quantity to be calculated recursively is the conditional PDF given the measurements outcome. Hence, a linear recursive filter can be interpreted as an approximated filter where the distribution of the state given the measurements is approximately Gaussian with mean $\hat{\mathbf{x}}^+$ and covariance matrix $\mathbf{P}^+_{\mathbf{xx}}$.

Experience has shown that the order of the statistical regression approximation of the covariance needs to be lower than that of the mean in order to obtain good numerical performance of the algorithm. A zeroth-order covariance approximation, therefore, has endured as a companion of a linear mean update rule, but it is also used in higher order update methodologies [8], [11], [23], [36], [38]. Our prior work, HOPUF-ℓ-c [38], presents a high-order polynomial state update, i.e., a higher-than-linear polynomial approximation of the conditional mean. This article presents a novel higher order polynomial covariance update to better approximate the conditional covariance than the standard zeroth-order approach.

A. The Polynomial Estimator

Gaussian filters are linear filters that approximate the distribution of the state given the measurements as Gaussian with mean $\hat{\mathbf{x}}^+$ and covariance matrix $\mathbf{P}^+_{\mathbf{xx}}$. This is equivalent to approximating the distribution of the state given the measurements as a linear transformation of a standard normal. This linear transformation is given by a shift of $\hat{\mathbf{x}}^+$ and a scale of $\sqrt{\mathbf{P}^+_{\mathbf{xx}}}$.

Our previous work (HOPUF-ℓ-c) expanded this concept by introducing a filter that approximates the distribution of the state given the measurements as a polynomial transformation of standard normal random variables and uses a higher-than-linear polynomial update function. This work introduces a novel covariance update technique and uses the HOPUF-ℓ-c state update, which is summarized here.

Let $\mathbf{x}$ be the state of the dynamic system, which is desired to be estimated, and let $\mathbf{y}$ be another random vector, sampleable, related to $\mathbf{x}$. Estimators are functions $\mathbf{g}(\mathbf{y})$ that infer the unknown value of $\mathbf{x}$ based on the known outcome of $\mathbf{y}$. Polynomial estimators are a subset of all estimators, which, using the Kronecker operator, can be written as

$$\mathbf{g}(\mathbf{y}) = \mathbf{a} + \mathbf{K}_1\mathbf{y} + \mathbf{K}_2\mathbf{y}^{[2]} + \mathbf{K}_3\mathbf{y}^{[3]} + \mathbf{K}_4\mathbf{y}^{[4]} + \cdots, \quad (3)$$

where $\mathbf{a}$ is a constant, each $\mathbf{K}_i$ is a constant matrix of appropriate dimensions, and each $\mathbf{y}^{[i]}$ is calculated using the Kronecker product

$$\mathbf{y}^{[i]} = \mathbf{y} \otimes \mathbf{y} \otimes \mathbf{y} \otimes \cdots. \quad (4)$$

In order to avoid redundancy, each repeated component of Equation (4) generated by the Kronecker product is eliminated, which means that, as an example, only one term between $y_i y_j$ and $y_j y_i$ is kept. It is convenient to derive the estimator's constants by working with deviation vectors. Deviation vectors are defined as

$$d\mathbf{x} = \mathbf{x} - \mathbb{E}\{\mathbf{x}\}, \quad (5)$$

$$d\mathbf{y}^{\{i\}} = \mathbf{y}^{[i]} - \mathbb{E}\left\{\mathbf{y}^{[i]}\right\}. \quad (6)$$

Deviations have zero mean by construction. The family of polynomial estimators defined by Equation (3) is redefined by adding and subtracting constants, in order to obtain a new, but theoretically equivalent, polynomial estimator family

$$\begin{aligned}\mathbf{g}(\mathbf{y}) &= \mathbf{a} + \mathbb{E}\{\mathbf{x}\} + \mathbf{K}_1(\mathbf{y} - \mathbb{E}\{\mathbf{y}\}) \\ &\quad + \mathbf{K}_2\left(\mathbf{y}^{[2]} - \mathbb{E}\left\{\mathbf{y}^{[2]}\right\}\right) \\ &\quad + \mathbf{K}_3\left(\mathbf{y}^{[3]} - \mathbb{E}\left\{\mathbf{y}^{[3]}\right\}\right) + \cdots \\ &= \mathbf{a} + \mathbb{E}\{\mathbf{x}\} + \mathbf{K}_1 d\mathbf{y} + \mathbf{K}_2\, d\mathbf{y}^{\{2\}} + \mathbf{K}_3\, d\mathbf{y}^{\{3\}} + \cdots \quad (7) \\ &= \mathbf{a} + \mathbb{E}\{\mathbf{x}\} + \mathcal{K} d\mathcal{Y}, \quad (8)\end{aligned}$$

where both the measurement residual with its powers, $d\mathcal{Y}$, and the matrices $\mathbf{K}_i$ are stacked:

$$\mathcal{K} = \begin{bmatrix}\mathbf{K}_1 & \mathbf{K}_2 & \mathbf{K}_3 & \ldots\end{bmatrix}, \quad (9)$$

$$d\mathcal{Y} = \begin{bmatrix}d\mathbf{y}^T & d\mathbf{y}^{\{2\}T} & d\mathbf{y}^{\{3\}T} & \ldots\end{bmatrix}^T. \quad (10)$$

The optimal estimator, in an MMSE sense, satisfies the orthogonality principle, from which it follows that the optimal polynomial update estimator becomes

$$\hat{\mathbf{x}} = \mathbb{E}\{\mathbf{x}\} + \mathbf{P}_{\mathbf{x}\mathcal{Y}}\mathbf{P}^{-1}_{\mathcal{Y}\mathcal{Y}} d\mathcal{Y}. \quad (11)$$

Matrices $\mathbf{P}_{\mathbf{x}\mathcal{Y}}$ and $\mathbf{P}_{\mathcal{Y}\mathcal{Y}}$ are the augmented state-measurement cross-covariance matrix and the augmented measurement covariance matrix, respectively. These matrices are constructed blockwise by using covariances $\mathbf{P}_{\mathbf{x}\mathbf{y}^{[j]}}$ and $\mathbf{P}_{\mathbf{y}^{[i]}\mathbf{y}^{[j]}}$, for any combination of i and j. As an example, $\mathbf{P}_{\mathbf{y}^{[3]}\mathbf{y}^{[4]}}$ indicates the covariance between the third-order measurement vector $\mathbf{y}^{[3]}$ and the fourth-order $\mathbf{y}^{[4]}$. Since deviations have zero mean by construction, the identities $\mathbf{P}_{\mathbf{y}^{[i]}\mathbf{y}^{[j]}} = \mathbf{P}_{d\mathbf{y}^{\{i\}}d\mathbf{y}^{\{j\}}}$ and $\mathbf{P}_{\mathbf{x}\mathbf{y}^{[j]}} = \mathbf{P}_{d\mathbf{x}d\mathbf{y}^{\{j\}}}$ are valid $\forall i, j \in \mathbb{N}_0$.

B. Differential Algebra

In this work, Gaussian random vectors undergo nonlinear (polynomial) transformations. The methodology used here to approximate these transformations is differential algebra (DA) via the Differential Algebra Core Engine (DACE2.0) software program. DA is used as a tool to implement the polynomial filter. Other approximations of nonlinear transformations are also possible but not considered here; ref. 23, for example, used the Unscented Transformation.

The theory of DA has been developed by Martin Berz in the late 1980s [7]. The DA framework is an algebra of Taylor polynomials. All functions are represented

through a matrix of coefficients and exponents rather than the classical representation with an array of floating point (FP) numbers. The DACE2.0 [27] software has a hard-coded library of the Taylor series expansion of elementary functions. As a consequence, derivatives are not computed numerically (e.g., finite differences), but evaluated directly from the Taylor polynomials. DA offers a way of working in a computer environment where the algebra of polynomials is endowed of composition of function, function inversions, explicit system solving, etc., as in the standard FP arithmetic.

DA has been proven to reduce computational costs in solving ordinary differential equations (ODEs) [28]. Once the maximum truncation order of the polynomial is selected, DA creates the Taylor polynomial expansion of the flow of ODEs as a function of the provided initial conditions. This approach can replace thousands of integrations with the computationally faster evaluation of the Taylor expansion [5]. As a result, the computational burden reduces considerably [42]. In the filtering problem, DA techniques have been used for the development of an efficient mapping of uncertainties [43] and for the evaluation of high-order moments [4]. Wittig *et al.* [45] developed a domain splitting technique that improves the state propagation when initial uncertainties are large by creating multiple polynomials.

The main concept of DA is that each function $f(\mathbf{x})$ can be expressed as a polynomial $p(\delta\mathbf{x})$, where the new variable $\delta\mathbf{x}$ is the deviation from the expansion center $\hat{\mathbf{x}}$. The polynomial $p(\delta\mathbf{x})$ is the Taylor series expansion of $f(\mathbf{x})$, centered at $\hat{\mathbf{x}}$, and truncated up to a user-selected order c.

For a detailed description of DA, its techniques, and how the DACE2.0 works in a computer environment, the reader is referred to the references.

III. THE STATE AND COVARIANCE ESTIMATION FILTER

A new filtering technique, based on a double polynomial estimator, is proposed in the DA framework. The double nature of the filter refers to the sequential estimation of the state and the covariance, where, at each time-step, the same measurement outcome is used twice to achieve matching between the conditioned state mean and its relative uncertainty spread.

Consider the generic dynamic system described by the following equations of motion and measurement equations:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{v}_k, \tag{12}$$

$$\mathbf{y}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}) + \mathbf{w}_{k+1}, \tag{13}$$

where $\mathbf{f}(\cdot)$ is the dynamics function, $\mathbf{x}_k$ is the n-dimensional state of the system at time-step k, $\mathbf{y}_{k+1}$ is the m-dimensional measurement vector at time-step $k+1$, and $\mathbf{h}(\cdot)$ is the measurement function. The noises are assumed to be zero-mean Gaussians and uncorrelated, such that their distribution is fully described by the first two moments. For all discrete time indexes i and j

$$\mathbb{E}\left\{\mathbf{v}_i\mathbf{w}_j^T\right\} = 0, \tag{14}$$

$$\mathbb{E}\left\{\mathbf{v}_i\mathbf{v}_j^T\right\} = \mathbf{Q}_i\delta_{ij}, \tag{15}$$

$$\mathbb{E}\left\{\mathbf{w}_i\mathbf{w}_j^T\right\} = \mathbf{R}_i\delta_{ij}, \tag{16}$$

where $\mathbf{Q}_i$ is the process noise autocovariance function, while $\mathbf{R}_i$ is for the measurement noise. The initial condition of the state of the system is assumed to be Gaussian as well $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$; however, for all other time-steps $k > 0$, the state distribution will be non-Gaussian due the nonlinearities in the dynamics.

The main result of this article, the State And Covariance Estimation Filter (SACE-c-η-μ), shares the prediction step with our previous work [38] and introduces a new update technique. The single distribution used in SACE-c-η-μ is expanded using Gaussian Multiple Models (GMMs) theory [30] to create the Multiple Models State And Covariance Estimation Filter (SACEMM-c-η-μ).

SACE-c-η-μ is composed of three different parts: the prediction, the state update, and the covariance update. The three integers c, η, and μ in SACE-c-η-μ refer to the tuning parameters of the filter: They are, respectively, the order of the Taylor polynomial approximation of $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$, (c), the order of the state polynomial update, (η), and the order of the covariance polynomial update, (μ).

A. Prediction

At the beginning of each time-step, the state distribution is assumed to be Gaussian $\mathbf{x}_k \sim \mathcal{N}(\hat{\mathbf{x}}_\mathbf{k}, \mathbf{P}_k)$. The state can, therefore, be initialized in the DA framework as a first-order polynomial

$$\mathbf{x}_k = \mathbf{x}_k(\delta\mathbf{x}_k) = \hat{\mathbf{x}}_\mathbf{k} + \mathbf{S}_k\delta\mathbf{x}_k, \tag{17}$$

where $\mathbf{S}_k\mathbf{S}_k^T = \mathbf{P}_k$, and the DA variable $\delta\mathbf{x}_k = \mathbf{x}_k - \hat{\mathbf{x}}_\mathbf{k}$ expresses the deviation from the expansion center, and it is interpreted as a Gaussian with zero mean and identity covariance matrix. Therefore, matrix $\mathbf{S}_k$ (here calculated through Cholesky Decomposition) scales the coefficients of the state polynomial and results in the moments of $\mathbf{x}_k$ easily calculated from the moments of $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

The propagation function is applied directly to the state polynomial, such that the predicted state vector is

$$\mathbf{x}_{k+1}^- = \mathbf{x}_{k+1}^-(\delta\mathbf{x}_k) = \mathbf{f}\big(\mathbf{x}_k(\delta\mathbf{x}_k)\big), \tag{18}$$

where $\mathbf{x}_{k+1}^-$ indicates the Taylor series expansion of the dynamics centered at $\hat{\mathbf{x}}_k$ and truncated at the user-defined integer order c. Equation (18) is carried out in

the DA framework. Each component of $\mathbf{x}_{k+1}^{-}$ is a polynomial map (centered at $\hat{\mathbf{x}}_k$) that maps deviations ($\delta\mathbf{x}_k$) from time-step k to time-step $k+1$ and describes how the state PDF evolves in time. The predicted polynomials are lacking the influence of the process noise. Process noise can be mapped in the DA framework with the same representation reserved for the state of the system. Thus, a new DA variable $\delta\mathbf{v}_k$, interpreted again as a standard normal random vector, is introduced:

$$\mathbf{x}_{k+1}^{-}(\delta\mathbf{x}_k, \delta\mathbf{v}_k) := \mathbf{x}_{k+1}^{-}(\delta\mathbf{x}_k) + \mathbf{T}_k\delta\mathbf{v}_k, \tag{19}$$

where $\mathbf{v}_k = \mathbf{T}_k\delta\mathbf{v}_k$ and $\mathbf{T}_k\mathbf{T}_k^T = \mathbf{Q}_k$.

Analogously, the predicted measurement is expressed as a Taylor polynomial expansion in the DA framework:

$$\mathbf{y}_{k+1} = \mathbf{y}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k) = \mathbf{h}\big(\mathbf{x}_{k+1}^{-}(\delta\mathbf{x}_k, \delta\mathbf{v}_k)\big), \tag{20}$$

where $\mathbf{y}_{k+1}$ is, again, a polynomial centered at $\hat{\mathbf{x}}_k$ with maximum order c. In Equation (20), the expansion is now with respect to both the state deviation vector ($\delta\mathbf{x}_k$) and the process noise ($\delta\mathbf{v}_k$). The influence of the measurement noise is added to the polynomials like in Equation (19). A new DA variable $\delta\mathbf{w}_{k+1}$ is introduced

$$\mathbf{y}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) := \mathbf{y}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k) + \mathbf{U}_{k+1}\delta\mathbf{w}_{k+1}, \tag{21}$$

where $\mathbf{w}_k = \mathbf{U}_k\delta\mathbf{w}_k$ and $\mathbf{U}_k\mathbf{U}_k^T = \mathbf{R}_k$. Once again, $\delta\mathbf{w}_{k+1}$ is interpreted as a standard normal random vector.

All the predicted quantities have been calculated, and they are represented as polynomial functions of standard random vectors. The number of variables is $2n+m$: n deviations map the state behavior, n map the process noise, and the remaining m map the measurement noise. The Gaussian nature of the random vectors leads to a fast evaluation of all expectation operations since, for a Gaussian PDF, central moments can be easily computed using Isserlis' formulation [17].

B. The State Polynomial Update

The second part of SACE-c-η-μ is the state polynomial update. After selecting the integer c in the prediction step, the user defines a second integer, η, which selects the order of the polynomial estimator dedicated to the state of the system.

The polynomial update evaluates the augmented Kalman gain and for high powers of the measurement polynomials. Starting from the latter,

$$\mathbf{y}_{k+1}^{[2]} = \mathbf{y}_{k+1} \otimes \mathbf{y}_{k+1}, \tag{22}$$

$$\mathbf{y}_{k+1}^{[i]} = \mathbf{y}_{k+1} \otimes \mathbf{y}_{k+1} \otimes \dots \tag{23}$$

with $i = 1, \dots, \eta$, and, once again, the redundant components are eliminated, in order to have independent measurements.

The means of the predicted state polynomials are now evaluated. Each polynomial undergoes the expectation operator, which, being a linear operator, works directly on the single monomials of the expansion [34].

Table I
Isserlis' Moments of Gaussian $\mathcal{N}(0, 1)$

Exponent	0	1	2	3	4	5	6	7	8	...
Coefficient	1	0	1	0	3	0	15	0	105	...

$$\hat{\mathbf{x}}^{-} = \mathbb{E}\left\{\mathbf{x}_{k+1}^{-}\right\}. \tag{24}$$

The deviations have a Gaussian distribution with zero mean and identity covariance matrix; therefore, the expected value substitutes the relative Isserlis' moment in for each monomial, according to Table I.

For example: $\mathbb{E}\left\{\alpha\delta x_1^8\delta x_2^4\delta x_4^6\delta v_2^2\delta w_3^4\right\} = 4725\alpha$. The predicted means of the measurement polynomials are similarly evaluated using Equation (24):

$$\hat{\mathbf{y}}_{k+1} = \mathbb{E}\left\{\mathbf{y}_{k+1}\right\}, \tag{25}$$

$$\hat{\mathbf{y}}_{k+1}^{[2]} = \mathbb{E}\left\{\mathbf{y}_{k+1}^{[2]}\right\}, \tag{26}$$

$$\hat{\mathbf{y}}_{k+1}^{[i]} = \mathbb{E}\left\{\mathbf{y}_{k+1}^{[i]}\right\}, \tag{27}$$

where, once again, $i = 1, \dots, \eta$.

The augmented measurement covariance $\mathbf{P}_{\mathcal{Y}\mathcal{Y},[\eta]}$ is evaluated blockwise. The matrix is guaranteed to be nonsingular because redundant rows and columns have been eliminated. The matrix is symmetric and each block is evaluated as

$$\mathbf{P}_{\mathbf{y}^{[i]}\mathbf{y}^{[j]}} = \mathbb{E}\left\{\left(\mathbf{y}_{k+1}^{[i]} - \hat{\mathbf{y}}_{k+1}^{[i]}\right)\left(\mathbf{y}_{k+1}^{[j]} - \hat{\mathbf{y}}_{k+1}^{[j]}\right)^{\mathrm{T}}\right\}, \tag{28}$$

$\forall i, j = 1, \dots, \eta$. Every time a polynomial multiplies itself, the maximum truncation order of the Taylor series doubles. For example, the evaluation of $\mathbf{P}_{\mathbf{y}^{[5]}\mathbf{y}^{[3]}}$ applies the expectation operator to a polynomial with monomials up to order $8c$. The augmented state-measurement cross-covariance matrix $\mathbf{P}_{\mathbf{x}\mathcal{Y},[\eta]}$ is evaluated blockwise, and each block is evaluated as

$$\mathbf{P}_{\mathbf{x}\mathbf{y}^{[i]}} = \mathbb{E}\left\{(\mathbf{x}_{k+1}^{-} - \hat{\mathbf{x}}_{k+1}^{-})\left(\mathbf{y}_{k+1}^{[i]} - \hat{\mathbf{y}}_{k+1}^{[i]}\right)^{\mathrm{T}}\right\}, \tag{29}$$

$\forall i = 1, \dots, \eta$. The subscript $[\eta]$ specifies that the covariance matrices are created with measurement powers up to order η. From these covariances, it is now possible to evaluate the augmented Kalman gain

$$\mathcal{K} = \mathbf{P}_{\mathbf{x}\mathcal{Y},[\eta]}\mathbf{P}_{\mathcal{Y}\mathcal{Y},[\eta]}^{-1}. \tag{30}$$

Denote with $\tilde{\mathbf{y}}_{k+1}$ the numerical outcome of the random vector $\mathbf{y}_{k+1}$, its powers are evaluated using the Kronecker product

$$\tilde{\mathbf{y}}_{k+1}^{[2]} = \tilde{\mathbf{y}}_{k+1} \otimes \tilde{\mathbf{y}}_{k+1}, \tag{31}$$

$$\tilde{\mathbf{y}}_{k+1}^{[i]} = \tilde{\mathbf{y}}_{k+1} \otimes \tilde{\mathbf{y}}_{k+1} \otimes \cdots \tag{32}$$

with $i = 1, \ldots, \eta$, and, once again, the redundant components are eliminated. The polynomial update exploits the influence of high powers from the measurement outcome. The measurement residual is developed to create the augmented innovation vector

$$d\tilde{\mathcal{Y}}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) = \begin{bmatrix} \tilde{\mathbf{y}}_{k+1} - \mathbf{y}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) \\ \tilde{\mathbf{y}}^{[2]}_{k+1} - \mathbf{y}^{[2]}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) \\ \cdots \\ \tilde{\mathbf{y}}^{[\eta]}_{k+1} - \mathbf{y}^{[\eta]}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) \end{bmatrix}. \tag{33}$$

The updated distribution (polynomial) of the state is given by

$$\begin{aligned} &\mathbf{x}^{+}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) \\ &\quad = \mathbf{x}^{-}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k) + \mathcal{K} d\tilde{\mathcal{Y}}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}), \end{aligned} \tag{34}$$

and the posterior estimate is its mean

$$\hat{\mathbf{x}}^{+}_{k+1} = \mathbb{E}\left\{\mathbf{x}^{+}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1})\right\} \tag{35}$$

evaluated, through Isserlis's moments, monomial by monomial using Table I.

Equation (34) shows that the state polynomials are a function of the three different deviations: the state deviation, the process noise, and the measurement noise. Furthermore, the new order of the polynomial is increased by a factor η, dictated by the order of the polynomial update. If the order of the polynomial approximation of the prior distribution ($\mathbf{x}^{-}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k)$) is c, then the order of the posterior polynomial ($\mathbf{x}^{+}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1})$) is ηc. The higher the polynomial order, the higher the number of moments to be calculated by Table I, which leads to a higher computational burden.

C. The Covariance Polynomial Update

The third, and last, part of SACE-c-η-μ is the covariance polynomial update. After having estimated the state of the system, SACE-c-η-μ applies a second polynomial estimator to identify the value of the state covariance conditioned to the measurements. Therefore, the user defines one last integer parameter, μ, that specifies the order of the covariance polynomial update. Unlike previous tuning parameters, μ cannot be freely chosen, but it has to respect the inequality $\mu < \eta$. The covariance cannot have an higher update order than the state.

The covariance matrix is obtained as

$$\mathbf{P}_{\mathbf{xx},k+1} = \mathbb{E}\left\{(\mathbf{x}^{+}_{k+1} - \hat{\mathbf{x}}^{+}_{k+1})(\mathbf{x}^{+}_{k+1} - \hat{\mathbf{x}}^{+}_{k+1})^{\mathrm{T}}\right\}. \tag{36}$$

This value shows the average spread of the posterior distribution among all different possible outcomes, $\tilde{\mathbf{y}}$, of the random variable $\mathbf{y}$. Equation (36) is the equivalent of the classical covariance update formulation, Equation (2), that is used in the most common filters such as EKF, UKF, QKF, CBF, Central Difference Filter, and GSOF. Therefore, even if correct, using the average error covariance does not extract all the possible information from the measurement outcome. Similar to the polynomial formulation for estimating the state presented in Equation (7), Equation (36) can be seen as a zeroth-order polynomial estimator of the covariance matrix.

A new approach is, therefore, presented in which the estimate of the covariance is performed to order higher than zero. Define a polynomial vector, $\boldsymbol{\rho}_{k+1}$, as the covariance polynomial

$$\boldsymbol{\rho}^{-}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) = (\mathbf{x}^{+}_{k+1} - \hat{\mathbf{x}}^{+}_{k+1}) \otimes (\mathbf{x}^{+}_{k+1} - \hat{\mathbf{x}}^{+}_{k+1}), \tag{37}$$

where, in order to reduce the computational burden, the redundant terms of the symmetric covariance matrix have been eliminated, e.g., the upper diagonal terms are removed. The covariance polynomial maximum order is $2\eta c$, being the square of the posterior distribution. The mean of $\boldsymbol{\rho}_{k+1}$ is exactly the vectorized version of the covariance matrix expressed in Equation (36):

$$\hat{\boldsymbol{\rho}}^{-}_{k+1} = \mathbb{E}\left\{\boldsymbol{\rho}^{-}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1})\right\}, \tag{38}$$

$$= \mathrm{stack}(\mathbf{P}_{\mathbf{xx},k+1}), \tag{39}$$

where the stack() operator indicates the vectorization of a matrix, performed by stacking columns on top of each other. The covariance update is treated in the same manner as the state vector: adding to a known prior a polynomial function of the measurement outcome $\tilde{\mathbf{y}}_{k+1}$. This second polynomial update provides an updated covariance value that better represents the state estimate's uncertainty.

The starting point is the already computed augmented measurement covariance matrix $\mathbf{P}_{\mathcal{Y}\mathcal{Y},[\mu]}$. The constrain $\mu < \eta$ makes $\mathbf{P}_{\mathcal{Y}\mathcal{Y},[\mu]}$ a subset of $\mathbf{P}_{\mathcal{Y}\mathcal{Y},[\eta]}$, obtained by selecting the first μ rows and columns. The cross-covariance matrix $\mathbf{P}_{\rho\mathcal{Y},[\mu]}$ is evaluated block-wise:

$$\mathbf{P}_{\rho\mathcal{Y},[\mu]} = \left[\mathbf{P}_{\rho\mathbf{y}}\ \mathbf{P}_{\rho\mathbf{y}^{[2]}}\ \mathbf{P}_{\rho\mathbf{y}^{[3]}}\ \cdots\right], \tag{40}$$

similarly to $\mathbf{P}_{\mathbf{x}\mathcal{Y},[\eta]}$. Each block is obtained as

$$\mathbf{P}_{\rho\mathbf{y}^{[i]}} = \mathbb{E}\left\{(\rho^{-}_{k+1} - \hat{\rho}^{-}_{k+1})\left(\mathbf{y}^{[i]}_{k+1} - \hat{\mathbf{y}}^{[i]}_{k+1}\right)^{\mathrm{T}}\right\} \tag{41}$$

with $i = 1, \ldots, \mu$. The Kalman gain associated with the covariance correction is calculated as

$$\mathcal{G} = \mathbf{P}_{\rho\mathcal{Y},[\mu]}\mathbf{P}^{-1}_{\mathcal{Y}\mathcal{Y},[\mu]}. \tag{42}$$

The covariance is updated to its posterior estimate as

$$\hat{\boldsymbol{\rho}}^{+}_{k+1} = \hat{\boldsymbol{\rho}}^{-}_{k+1} + \mathcal{G}\begin{bmatrix} \tilde{\mathbf{y}}_{k+1} - \hat{\mathbf{y}}_{k+1} \\ \tilde{\mathbf{y}}^{[2]}_{k+1} - \hat{\mathbf{y}}^{[2]}_{k+1} \\ \cdots \\ \tilde{\mathbf{y}}^{[\mu]}_{k+1} - \hat{\mathbf{y}}^{[\mu]}_{k+1} \end{bmatrix}, \tag{43}$$

where the influence of the measurement is weighted by the augmented Kalman gain. Before starting the next iteration, vector $\hat{\boldsymbol{\rho}}^{+}_{k+1}$ is brought back to its matrix formulation

$$\hat{\mathbf{P}}_{\mathbf{xx},k+1} = \mathrm{matrix}(\hat{\boldsymbol{\rho}}^{+}_{k+1}), \tag{44}$$

where the matrix() operator is the inverse of the stack() operator.

The updated posterior distribution can be approximated as Gaussian with mean $\hat{\mathbf{x}}^{+}_{k+1}$ and covariance matrix $\hat{\mathbf{P}}_{\mathbf{xx},k+1}$ to start the next iteration from Equation (17), where the DA variables related to the noises are discarded and a new state deviation vector is initialized.

SACE-c-η-μ contains three tuning parameters to enhance the performance of classic estimators. In fact, SACE-1-1-0 reduces to the EKF, and SACE-2-1-0 is the GSOF. The polynomial estimator better weights the information from the measurement by computing high-order central moments. The increase in accuracy is paid by an increase in computational effort, which practically limits the filter's order selection. The highest polynomial order the filter has to compute (in the evaluation of $\mathbf{P}_{\rho\mathbf{y}^{[\mu]}}$) is $(2\eta+\mu)c$.

The computational time required by the filter depends on the selection of its three tuning parameters and on the dimension of the state vector. SACE-c-η-μ is not suitable for extremely large systems because of the exponential grow in the number of monomials in the Taylor expansion [9]. An in-depth analysis of the computational time of filters developed in the DACE2.0 framework is presented in [12]. The reference portraits an exhaustive analysis of the execution time on the BeagleBone Black (BBB) Single Board Computer, with particular focus on the duty cycles of filter execution on BBB and its dependency on the Taylor truncation order.

IV. THE MULTIPLE MODELS SPACE AND COVARIANCE ESTIMATION FILTER

SACE-c-η-μ approximates the time propagation of the state with one single polynomial representation of the flow. However, as the Taylor polynomial series gets farther away from the expansion center, it becomes less accurate. Therefore, when the initial uncertainties of the state distribution are extremely large, a single polynomial map may not be sufficient to truthfully describe the predicted PDF [45]. Splitting the initial uncertainties in multiple (smaller) subdomains aids the filter in reaching convergence. Thus, a second filter called SACEMM-c-η-μ merges SACE-c-η-μ with the GMM formulation. In the DA framework, multiple models translate into multiple polynomials.

A. Initialization

The initial state distribution is assumed to be Gaussian $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_\mathbf{0}, \mathbf{P}_0)$. The initialization of the models follows an analogy with the unscented transformation [20]. Therefore, the initial domain is divided into $\theta = 2n+1$ models, where n is the number of states. Each ith model is a Gaussian with mean $\hat{\mathbf{x}}_{0,\{i\}}$ and covariance $\mathbf{P}_{0,\{i\}}$. Being symmetric, the state covariance matrix can be elaborated into its eigenvalue decomposition

$$\mathbf{P}_0 = \mathbf{V}\mathbf{D}\mathbf{V}^T, \tag{45}$$

where $\mathbf{V}$ is the matrix of eigenvectors that describes the orientation of the uncertainty ellipsoids, and the diagonal matrix of eigenvalues $\mathbf{D}$ describes the magnitude of the uncertainties. The mean of each Gaussian kernel is selected as

$$\hat{\mathbf{x}}_{0,\{0\}} = \hat{\mathbf{x}}_0, \tag{46}$$

$$\hat{\mathbf{x}}_{0,\{j\}} = \hat{\mathbf{x}}_0 + \mathbf{V}\mathbf{D}_j, \qquad j = 1,\dots,n, \tag{47}$$

$$\hat{\mathbf{x}}_{0,\{j\}} = \hat{\mathbf{x}}_0 + \mathbf{V}\mathbf{D}_{j-n}, \quad j = n+1,\dots,2n, \tag{48}$$

where $\mathbf{D}_j$ indicates the jth column of the matrix. The centers of the models lie on the principal axes and their initial weights are proportional to their probability with respect to the initial distribution

$$\omega_{0,\{i\}} = \frac{(2\pi)^{-n/2}}{\mathcal{W}_0\sqrt{\det\mathbf{P}_0}} \exp\left(-\frac{1}{2}\left(\hat{\mathbf{x}}_{0,\{i\}}-\hat{\mathbf{x}}_0\right)^T\mathbf{P}_0^{-1}\left(\hat{\mathbf{x}}_{0,\{i\}}-\hat{\mathbf{x}}_0\right)\right), \tag{49}$$

$$\mathcal{W}_0 = \sum_{i=0}^{\theta-1}\omega_{0,\{i\}}, \tag{50}$$

where $\mathcal{W}_0$ normalizes the weights such that their sum is unity. The models are assumed to share the same covariance, and they all have the same initial level of uncertainty

$$\mathbf{P}_{0,\{j\}} = \mathbf{P}_0 + \mathbf{x}_0\mathbf{x}_0^T - \sum_{i=0}^{\theta-1}\omega_{0,\{i\}}\mathbf{x}_{0,\{i\}}\mathbf{x}_{0,\{i\}}^T \tag{51}$$

with $j = 0,\dots,\theta-1$. Therefore, at the beginning of the first iteration, the initial Gaussian distribution has been divided into θ smaller Gaussian kernels $\mathbf{x}_{0,\{i\}} \sim \mathcal{N}(\hat{\mathbf{x}}_{0,\{i\}}, \mathbf{P}_{0,\{i\}})$ with the same covariance matrix and means on the principal axes, selected as sigma points from the unscented transformation.

B. Prediction

The models have been initialized as Gaussian kernels. SACEMM-c-η-μ applies SACE-c-η-μ on each kernel like it were operating by its own. As a consequence, θ different polynomials are created in the DA frameworks, and θ polynomial maps of the flow describe the time propagation of the state.

$$\mathbf{P}_{k,\{i\}} = \mathbf{S}_{k,\{i\}}\mathbf{S}_{k,\{i\}}^T, \tag{52}$$

$$\delta\mathbf{x}_{k,\{i\}} = \mathbf{x}_{k,\{i\}} - \hat{\mathbf{x}}_{k,\{i\}}, \tag{53}$$

$$\mathbf{x}_{k,\{i\}} = \mathbf{x}_{k,\{i\}}(\delta\mathbf{x}_{k,\{i\}}) = \hat{\mathbf{x}}_{k,\{i\}} + \mathbf{S}_{k,\{i\}}\delta\mathbf{x}_{k,\{i\}}, \tag{54}$$

$$\mathbf{x}^{-}_{k+1,\{i\}} = \mathbf{x}^{-}_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}) = \mathbf{f}\big(\mathbf{x}_{k,\{i\}}(\delta\mathbf{x}_{k,\{i\}})\big), \tag{55}$$

with $i = 0,\dots,\theta-1$. Multiple Taylor series expansions improve the approximation accuracy of the polynomial

maps since, at the boundaries, deviations are closer to their relative centers. Following SACE-c-η-μ for each model, the process noise is mapped on each polynomial expansion:

$$\mathbf{v}_{k,\{i\}} = \mathbf{T}_k \delta\mathbf{v}_{k,\{i\}}, \tag{56}$$

$$\mathbf{x}^-_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}, \delta\mathbf{v}_{k,\{i\}}) := \mathbf{x}^-_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}) + \mathbf{T}_k \delta\mathbf{v}_{k,\{i\}}, \tag{57}$$

and a measurement polynomial is evaluated for each kernel:

$$\mathbf{w}_{k+1,\{i\}} = \mathbf{U}_{k+1} \delta\mathbf{w}_{k+1,\{i\}}, \tag{58}$$

$$\begin{aligned}\mathbf{y}_{k+1,\{i\}} &= \mathbf{y}_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}, \delta\mathbf{v}_{k,\{i\}}),\\ &= \mathbf{h}\big(\mathbf{x}^-_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}, \delta\mathbf{v}_{k,\{i\}})\big).\end{aligned} \tag{59}$$

Measurement noise is added in the DA framework

$$\begin{aligned}&\mathbf{y}_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}, \delta\mathbf{v}_{k,\{i\}}, \delta\mathbf{w}_{k+1,\{i\}}) :=\\ &\quad \mathbf{y}_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}, \delta\mathbf{v}_{k,\{i\}}) + \mathbf{U}_{k+1}\delta\mathbf{w}_{k+1,\{i\}}\end{aligned} \tag{60}$$

such that the prediction step is completed for each Gaussian kernel.

C. The State and Covariance Polynomial Update

The prediction step has been exploited by the introduction of multiple polynomials. In the update step, each kernel undergoes the polynomial update for the state and for the covariance described by SACE-c-η-μ. Therefore, after having selected η and μ as the orders for the polynomial estimators, the state posterior estimate and the conditional covariance of each model are evaluated as

$$\mathbf{x}^+_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}, \delta\mathbf{v}_{k,\{i\}}, \delta\mathbf{w}_{k+1,\{i\}}) = \mathbf{x}^-_{k+1,\{i\}} + \mathcal{K}_{\{i\}} d\tilde{\mathcal{Y}}_{\{i\}}, \tag{61}$$

$$\hat{\mathbf{x}}^+_{k+1,\{i\}} = \mathbb{E}\left\{\mathbf{x}^+_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}, \delta\mathbf{v}_{k,\{i\}}, \delta\mathbf{w}_{k+1,\{i\}}\right\}, \tag{62}$$

and

$$\hat{\boldsymbol{\rho}}^+_{k+1,\{i\}} = \hat{\boldsymbol{\rho}}^-_{k+1,\{i\}} + \mathcal{G}_{\{i\}} \begin{bmatrix} \tilde{\mathbf{y}}_{k+1,\{i\}} - \hat{\mathbf{y}}_{k+1,\{i\}} \\ \tilde{\mathbf{y}}^{[2]}_{k+1,\{i\}} - \hat{\mathbf{y}}^{[2]}_{k+1,\{i\}} \\ \cdots \\ \tilde{\mathbf{y}}^{[\mu]}_{k+1,\{i\}} - \hat{\mathbf{y}}^{[\mu]}_{k+1,\{i\}} \end{bmatrix}, \tag{63}$$

$$\hat{\mathbf{P}}_{\mathbf{xx},k+1,\{i\}} = \text{matrix}(\hat{\boldsymbol{\rho}}^+_{k+1,\{i\}}), \tag{64}$$

with $i = 0, \ldots, \theta - 1$. Every Kalman gain and expectation has been calculated according to the polynomial estimator theory and using Table I, since each deviation is interpreted as a standard normal random vector.

The influence of each ith Gaussian to the posterior PDF needs to be updated as well. The posterior distribution of the probability of each Gaussian given the measurements can be evaluated using Bayes' rule. Therefore, the updated weight of each model is proportional to its measurement likelihood. Let us define with $P(\mathbf{y}_{k+1}|i, \mathbf{Y}_k)$ the probability of $\tilde{\mathbf{y}}_{k+1}$ to be the outcome from the ith Gaussian:

$$\begin{aligned}P(\tilde{\mathbf{y}}_{k+1}|i, \mathbf{Y}_k) &= \frac{(2\pi)^{-m/2}}{\sqrt{\det \mathbf{P}_{\mathbf{yy},\{i\}}}}\\ &\exp\left(-\frac{1}{2}\left(\tilde{\mathbf{y}}_{k+1} - \hat{\mathbf{y}}_{k+1,\{i\}}\right)\mathbf{P}^{-1}_{\mathbf{yy},\{i\}}\left(\tilde{\mathbf{y}}_{k+1} - \hat{\mathbf{y}}_{k+1,\{i\}}\right)\right),\end{aligned} \tag{65}$$

where $\mathbf{Y}_k$ indicates all the measurement realizations up to time-step k. The weight update formulation is derived, for the ith kernel, as

$$\begin{aligned}\omega_{k+1,\{i\}} &= P(i|\mathbf{Y}_{k+1})\\ &= P(i|\tilde{\mathbf{y}}_{k+1}, \mathbf{Y}_k)\\ &= \frac{P(i, \tilde{\mathbf{y}}_{k+1}|\mathbf{Y}_k)}{P(\tilde{\mathbf{y}}_{k+1}|\mathbf{Y}_k)}\\ &= \frac{P(i, \tilde{\mathbf{y}}_{k+1}|\mathbf{Y}_k)}{\sum_{j=0}^{\theta-1} P(j, \tilde{\mathbf{y}}_{k+1}|\mathbf{Y}_k)}\\ &= \frac{P(\tilde{\mathbf{y}}_{k+1}|i, \mathbf{Y}_k)P(i|\mathbf{Y}_k)}{\sum_{j=0}^{\theta-1} P(j, \tilde{\mathbf{y}}_{k+1}|\mathbf{Y}_k)}\\ &= \frac{P(\tilde{\mathbf{y}}_{k+1}|i, \mathbf{Y}_k)}{\sum_{j=0}^{\theta-1} \omega_{k,\{j\}} P(\tilde{\mathbf{y}}_{k+1}|j, \mathbf{Y}_k)}\omega_{k,\{i\}},\end{aligned} \tag{66}$$

where the denominator normalizes the weights such that they sum to unity. Equation (66) is recursive and modifies the importance of each model based on how likelihood it could have generated the measurement outcome.

The filtering algorithm has ended, and it can start the following iteration from $\hat{\mathbf{x}}^+_{k+1,\{i\}}$, $\hat{\mathbf{P}}_{\mathbf{xx},k+1,\{i\}}$, and $\omega_{k+1,\{i\}}$ for each model. However, the weighted state estimate, $\bar{\mathbf{x}}$, and covariance, $\bar{\mathbf{P}}$, are calculated for downstream users, and they are used to assess the performance of the filtering technique.

$$\bar{\mathbf{x}} = \sum_{i=0}^{\theta} \omega_{k+1,\{i\}} \hat{\mathbf{x}}^+_{k+1,\{i\}}, \tag{67}$$

$$\bar{\mathbf{P}} = -\bar{\mathbf{x}}\bar{\mathbf{x}}^T + \sum_{i=0}^{\theta} \omega_{k+1,\{i\}} \left(\hat{\mathbf{P}}_{\mathbf{xx},k+1,\{i\}} + \hat{\mathbf{x}}^+_{k+1,\{i\}}\hat{\mathbf{x}}^{+T}_{k+1,\{i\}}\right). \tag{68}$$

Once again, for basic parameters, SACEMM-c-η-μ reduces to well-known filters: In fact, picking SACEMM-1-1-0 reduces to the GSF. The computational complexity of SACEMM-c-η-μ is approximately θ times bigger when compared to SACE-c-η-μ. Therefore, it is advised to operate the multiple-model technique when the initial state uncertainties are particularly large, or when the time-step is long enough that one polynomial approximation is not sufficient to adequately represent the flow of the dynamics. Therefore, for problems with high initial uncertainty, SACEMM-c-η-μ can be used for the first few iteration steps and then replaced with SACE-c-η-μ once the sate error covariance has decreased.

V. NUMERICAL EXAMPLES

The proposed filtering techniques have been applied to three different scenarios. First, a scalar application gives a visual representation of how the new update algorithm works and highlights the innovative features as compared to other estimators. The second problem consists of a tracking application where the system undergoes the highly nonlinear dynamics of a Lorenz96 system. The third application uses Lorenz63 dynamics to underline the benefits of the multiple model filtering technique.

A. Scalar Problem

A simple scalar problem is presented here to highlight the improvements of the new filtering technique by estimating the conditional covariance. It has already been proven that high-order polynomial estimators are a better approximation of the true MMSE [38]. However, the presented example underlines the matching between state and covariance for each different realization of the measurement.

Define a normal prior state distribution $x \sim \mathcal{N}(1, 0.02)$ and a measurement

$$y = 1/x + \nu, \tag{69}$$

where $\nu \sim \mathcal{N}(0, 0.003)$ is independent of x and represents the measurement noise.

Fig. 1 shows the true joint distribution of x and y represented using 10^5 points (gray dots in the figure). The figure compares SACE-c-η-μ and SACEMM-c-η-μ with a few common estimators: the EKF, the UKF, the GSF, the Iterated Extended Kalamn Filter (IEKF) [6], the PF, and the high-order EKF (DAHO-k) [43]. The first row of graphs (EKF, UKF, DAHO-3) contains linear estimators; therefore, their representation on the (x, y) plane is a straight line, shown in red. The slope of the red line is the Kalman gain, whose optimal value is $\mathbf{P_{xy}}\mathbf{P_{yy}}^{-1}$. The different slopes shown by the different linear estimators

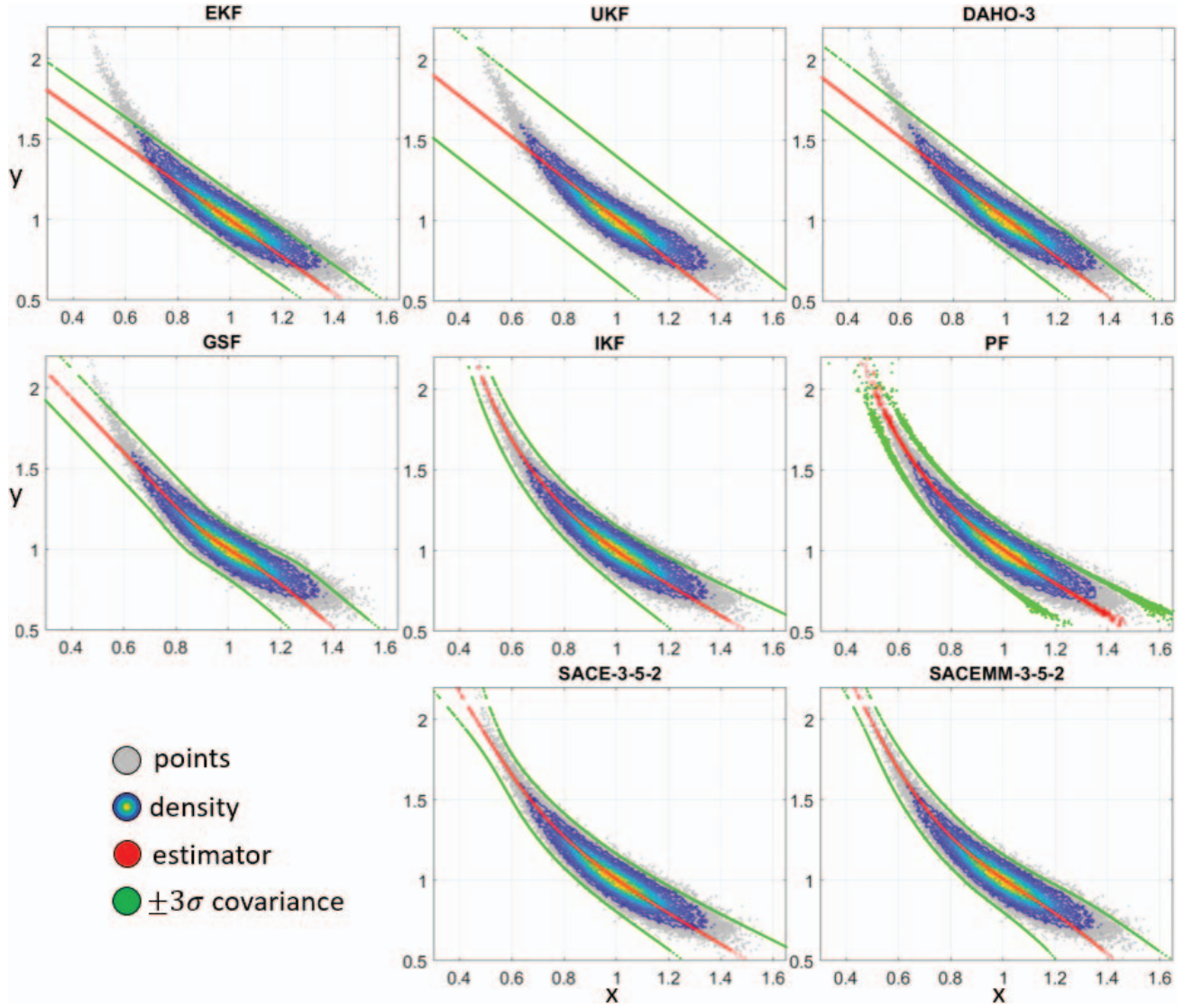


Fig. 1. Comparison among different estimators. Posterior distribution (gray), the estimator functions (red), and their confidence levels (green).

are due to the different approximations each linear filter employs to evaluate the moments. The EKF applies basic linearization (Jacobians), the UKF uses the unscented transformation, and DAHO-3 uses Taylor polynomials up to the third order. The green lines depict the filter's own assessment of the estimation error uncertainty as a $\pm 3\sigma$ boundary. The different evaluations of the moments lead to different values on the estimation of the variance, as it follows Equation (2). The green lines share the same slope of the corresponding red line: They are just translated left (and right) by 3σ. These linear filters estimate the same uncertainty level regardless of the measurement outcome, and the predicted covariance value is the mean among all the possible different realizations.

The second row of graphs in Fig. 1 shows nonlinear estimators. The GSF has been implemented with three models, which allows the estimator function, red line, to follow the curved shape of the posterior distribution. However, when the likelihood of one model becomes predominant with respect to the others, the GSF behaves similarly to the EKF: This aspect is mostly evident near the tails of the distribution. The estimated covariance of the GSF is a function of the measurement because it is evaluated as a weighted mean among all the models, whose importance weight is based on their likelihood. However, the $\pm 3\sigma$ green lines show the same problems of linear estimators: The lines are able to change slope when the models have approximately the same weight; otherwise, they are straight. Furthermore, since the GSF can be intended as multiple EKFs with reduced subdomains, the filter shows the same behavior of the linear estimator at the edges of the posterior PDF. The IEKF performs multiple updates to repeatedly calculate the measurement Jacobian each time linearizing with respect to the most current estimate. The IEKF minimizes a nonlinear least-squares performance index that, for appropriate probability distributions functions, approximates the maximum *a posteriori* (MAP) estimate. As such, the IEKF is a nonlinear estimator, and its red line follows the bend of the posterior distribution, setting on the most likely value of x for each measurement outcome y. The $\pm 3\sigma$ green lines correctly bound the distribution; however, the IEKF is not necessary an unbiased filter, and choosing the peak of the posterior distribution does not necessarily minimize the MSE. Hence, the IEKF's MSE is often larger than filters based on the MMSE principle [37]. The third nonlinear estimator presented in Fig. 1 is the PF. PFs are accurate nonlinear estimators that use an ensemble of weighted particles to calculate the state estimate. The weight of each particle depends on its measurement likelihood. Both the state estimate and the predicted error covariance are (nonlinear) functions of the measurements. The graph shows that the PF estimates do not form well-defined lines, but the state and covariance estimate values depend on the randomness of the data. In other words, while in the EKF the state estimates from two separate updates with the same measurement outcome give exactly the same value, two PF estimates depend on the randomness of the initial ensemble used to generate them. Consequently, the green and red "lines" of the PF become thicker while moving towards the tails of the posterior distribution.

In the third row, SACE-3-5-2 and SACEMM-3-5-2 are reported. The fifth-order polynomial estimator is able to follow the curved shape of the joint distribution, and it accurately approximates the true MMSE. The optimal MMSE is the conditional mean, which visually is the line that divides in half the distribution of y, as horizontal spread of points, for each value of x. Therefore, while EKF, UKF, and DAHO-3 can be interpreted as different linear approximations of the true MMSE, SACE-3-5-2 represents a fifth-order approximation, which shows a more accurate result. By increasing the estimator order η to infinity, SACE-c-η-μ would asymptotically reach the true MMSE. The green lines related to SACE-3-5-2 show how the uncertainty level has become a (nonlinear) function of the measurement. The $\pm 3\sigma$ boundary increases and tightens depending on the horizontal spread of samples around the estimator function. For example, when the current measurement is $y = 1$, SACE-3-5-2 gives its estimate with a level of uncertainty that matches the spread of the gray points on the line $y = 1$. When the sensor gives $y = 2$, SACE-3-5-2 outputs a level of confidence in its estimate higher than in the previous case, since the spread of the gray samples around its estimate at $y = 2$ is tighter. Therefore, the estimated covariance of the filter is a function of the measurement, and the performance improves drastically because the uncertainty level always matches the estimate, providing a more reliable outcome. There appear to be no influential benefits in applying the multiple model polynomial estimator: SACEMM-3-5-2 behaves similarly to its single-model counterpart and shares the same features. However, at the tails of the distribution, SACEMM-3-5-2 estimated conditional covariance better follows the distribution of the samples.

The accuracy level reached by each filter is compared in Fig. 2, where the results of a RMSE analysis is reported.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N_{\mathrm{samples}}} (x_i - \hat{x}_i^+)^2}{N_{\mathrm{samples}}}}. \tag{70}$$

The RMSE of each estimator is evaluated using the entire set of 10^5 points. The bars show that SACE-3-5-2 is the most accurate filter, while the linear estimators are the least. However, a more precise approximation of the measurement equation leads to a smaller RMSE and to a more precise estimate, as proven by DAHO-3 (third-order Taylor polynomial) being the most accurate among the other linear estimators. The IEKF shares the same accuracy level as DAHO-3, while the other nonlinear estimators have lower RMSE. Two PF implementations are shown with different numbers of particles: 10^3

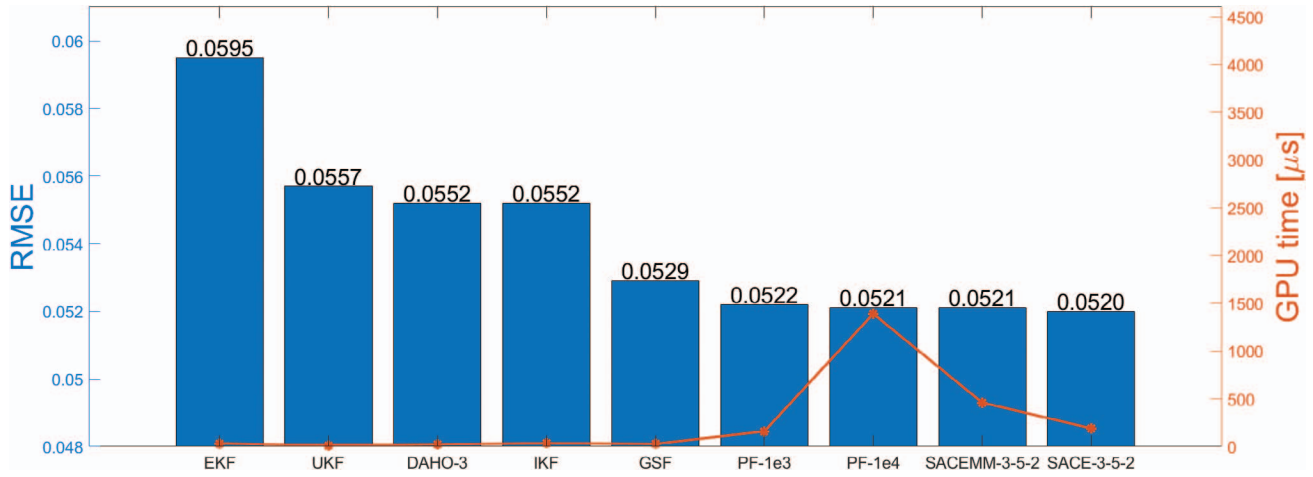


Fig. 2. Comparison of RMSE and computational time among different estimators.

and 10^4. It has error levels comparable with SACE-3-5-2, and the PF with 10^4 particles has a heavier computational burden. Fig. 2 reports, in orange, the average GPU time of each estimator, evaluated among all the 10^5 runs shown in Fig. 1. As expected, PF-1e4 has the highest computational time, while linear estimators have the lowest. SACE-3-5-2 achieves the best accuracy levels comparable to sample-based filters in a shorter amount of time, although the performance of all nonlinear filters is very similar in this simple motivating example.

The proposed scalar problem shows no significant difference between SACE-c-η-μ and SACEMM-c-η-μ. Let us increase the prior uncertainty level to $x \sim \mathcal{N}(1, 0.03)$ in order to underline the benefits of having multiple models. Fig. 3 shows the estimator function and confidence level of the two filters, along with the joint distribution. SACE-3-5-2 outputs unphysical results for the predicted conditional covariance of the state: a negative value of σ^2. The initial prior uncertainties are excessively large to allow the filter to work properly. On the left tail of the join distribution, the variance becomes negative and that is represented by the green lines overlapping the red one, to show that the filtering algorithm is not functioning correctly. SACEMM-3-5-2, on the other hand, has no issues in estimating correctly both the state and the covariance for all possible outcomes of the measurement. The green lines bound the samples of the joint distribution narrowing and widening as needed. The correct result is connected to the reduced initial covariance associated with each model, which increases the filter robustness and performance.

The proposed problem underlines a couple of characteristics of the proposed algorithms. Unlike the linear and Gaussian cases, the conditional covariance and the estimation error covariance are different. Linear filters employ the estimation error covariance, which expresses the average spread of the estimation error over all possible measurement realizations. This is a good metric, but once a measurement is actually available to process, the covariance conditioned on the actual measurement outcome is a more informative quantity, because it provides the spread of the estimation error for the actual value of y. In fact, the conditional covariance is a (nonlinear) function of the measurement whose evaluation is usually not feasible. SACE-c-η-μ and SACEMM-c-η-μ use a polynomial estimator to approximate the function, achieving better results with respect to filters that do not.

B. Lorenz96 System

The performance of the proposed filter is tested on a Lorenz96 example [30], where the state dynamics are

$$\frac{dx_i(t)}{dt} = x_{i-1}(t)(x_{i+1}(t) - x_{i-2}(t)) - x_i(t) + F + v_i(t), \tag{71}$$

with $i = 1, \ldots, 4$, since $\mathbf{x}(t)$ is selected to be four-dimensional. The following conventions are used: $x_{-1}(t) = x_{n-1}(t)$, $x_0(t) = x_n(t)$, and $x_1(t) = x_{n+1}(t)$. The term F is a constant external force with value chosen equal to 8, since it introduces a chaotic behavior in the system. The initial condition is assumed to be Gaussian, with mean $\hat{\mathbf{x}} = \begin{bmatrix} F & F & F + 0.01 & F \end{bmatrix}^T$ and diagonal covariance matrix, with the same standard deviation for each component of the state: $\sigma_{\mathbf{x}} = 10^{-3}$. The process noise is assumed to be Gaussian and uncorrelated among states, with known standard deviation $\sigma_v = 10^{-3}$. The dynamics are propagated at 2 Hz for a total of 20 s. The measurements are obtained at each time-step according to the following model:

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}(t_k) + \boldsymbol{\mu}_k, \quad \mathbf{H}_{i,j} = \begin{cases} 1 & j = 2i - 1 \\ 0 & \text{otherwise} \end{cases} \tag{72}$$

with $i = \{1, 2\}$ and $j = \{1, 2, 3, 4\}$. In other words, the sensors observe the components of the state with odd indices. Measurement noises are assumed to be Gaussian and uncorrelated within each other and with the process noise. The standard deviation is selected as $\sigma_{\boldsymbol{\mu}} = 0.5$: This value is particularly high, and filters based on linear estimators are not able to track the state of the system and achieve convergence [38].

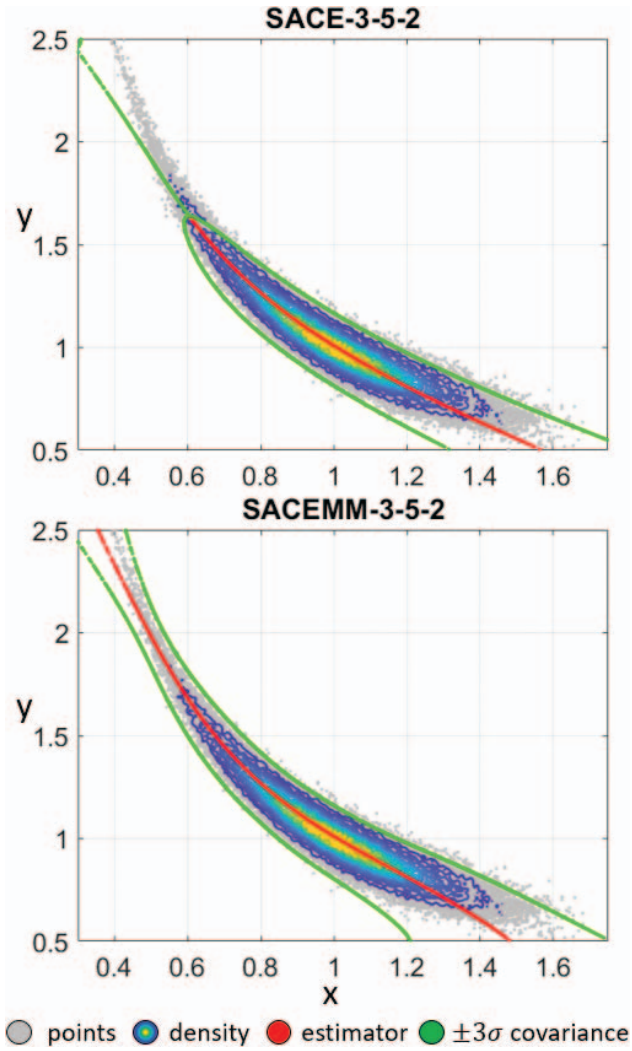


Fig. 3. SACE-3-5-2 versus SACEMM-3-5-2. Posterior distribution (gray), the estimator functions (red), and their confidence levels (green).

Fig. 4 shows the Monte Carlo analysis results performed with SACE-2-3-2 on the presented application. The figure shows, for each ith component of the state, the estimation error of each realization (gray lines), calculated as

$$\epsilon_{j,i} = x_{j,i} - \hat{x}_{j,i}, \tag{73}$$

for each jth time-step. A total of 100 realizations are reported. Fig. 4 describes the error means, in black, and the error standard deviations, as 3σ values, in blue. The black lines show that SACE-c-η-μ is an unbiased filter, as expected from the theory of MMSE estimation. The predicted error standard deviation, continuous blue line, is evaluated directly from the updated covariance matrix, by taking the square root of the diagonal terms. The effective performance of the filter is assessed by the sample standard deviation of the Monte Carlo estimation errors, dashed blue lines. At each time-step, the actual error covariance of the filter is evaluated by working directly on the samples. The consistency of SACE-2-3-2 is established by the overlapping of the dashed and continuous blue lines, which proves that the filter can correctly predict its own uncertainty levels.

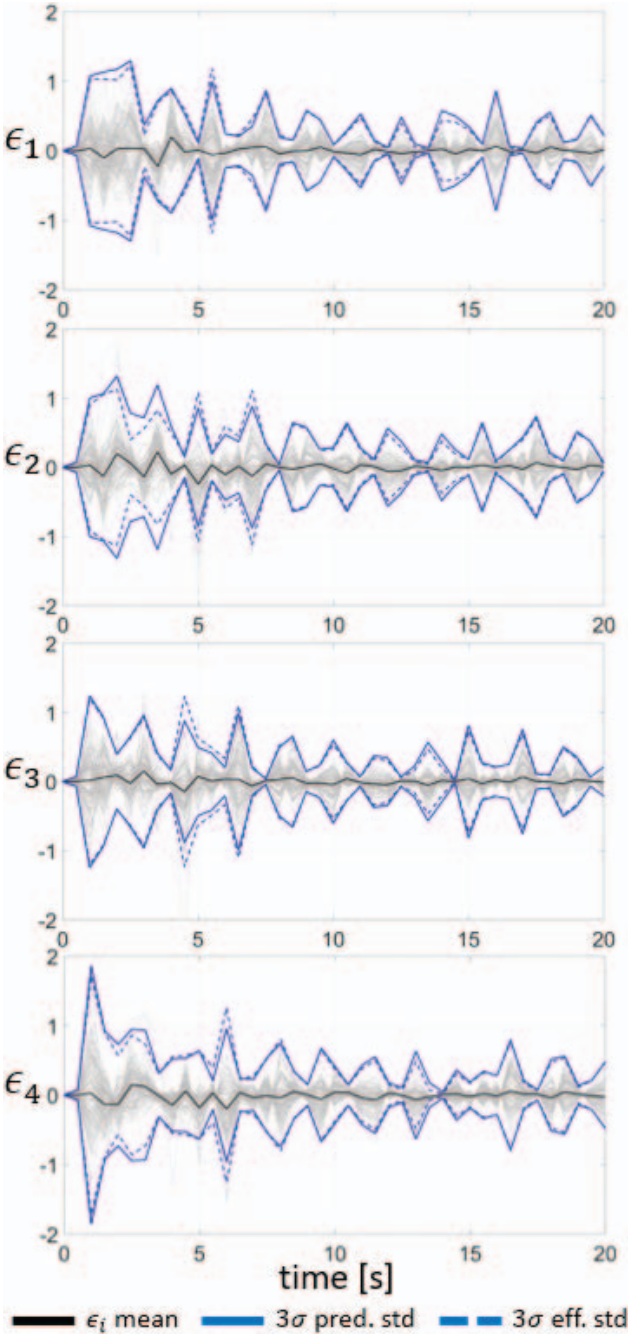


Fig. 4. Monte Carlo performance analysis with SACE-2-3-2: 100 runs.

The performance comparison among different filters is shown in Fig. 5 through another Monte Carlo analysis conducted with 100 runs. The figure shows, for each filter, the comparison between the effective and predicted error covariance. The continuous lines represent the filter's own estimate of the error standard deviation, calculated directly from the updated covariance matrix as the square root of its trace:

$$\bar{\sigma} = \sqrt{\mathrm{tr}(\hat{\mathbf{P}}_{\mathbf{xx}})}. \tag{74}$$

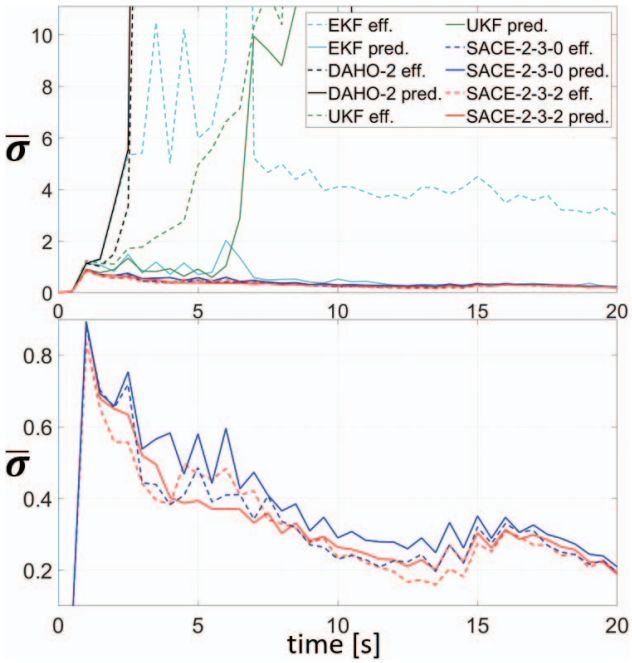


Fig. 5. Lorenz96: covariance comparison among different filters.

The dashed lines represent the effective error standard deviation derived from the Monte Carlo analysis. A consistent filter has the matching between its dashed and continuous lines, meaning that the estimated uncertainty level reflects the actual error standard deviation. The top graph, in Fig. 5 shows how linear estimators, the EKF, UKF, and DAHO-2, diverge (and break down) while trying to track the state of the system. The measurement noise level is excessively large, and a linear dependence on the measurement outcome is not sufficient to achieve a correct estimate. The EKF lines also represent the behavior of the IEKF: Since the measurement is a linear measurement, the IEKF reduces to the EKF. The UKF and DAHO-2 use, respectively, the unscented transformation and second-order Taylor polynomial to improve the prediction step of the filter and have a more accurate propagated state prior distribution. However, the update step is still linear and highly influenced by the noise standard deviation that prevents the evaluation of a reliable Kalman gain. The polynomial estimator better weights the information from the measurements using high-order moments and it achieves convergence and consistency. Therefore, SACE-2-3-0, in blue, and SACE-2-3-2, in red, correctly estimate the state of the system along the whole simulation. The bottom graph in Fig. 5 zooms in on the performance of SACE-c-η-μ for the two different sets of parameters. SACE-2-3-0 shows a filter whose estimate is a polynomial function of the measurement, and its estimated covariance is evaluated as a mean among all possible resolutions; it is not influenced by the measurement outcome. SACE-2-3-2, on the other hand, improves accuracy by estimating the covariance, giving it the same importance reserved for the state. Thus, the red lines settle below the blue ones for the whole simulation, since the predicted error standard deviation better matches the conditional mean.

C. Lorenz63 System

The performance of the proposed algorithms is also tested on a Lorenz63 application [30], [41], a challenging nonlinear system without process noise. The absence of process noise causes impoverishments in PFs, typically resulting in unsatisfactory performance. The state of the system undergoes the following dynamics:

$$\frac{dx_1(t)}{dt} = \alpha(x_2(t) - x_1(t)), \tag{75}$$

$$\frac{dx_2(t)}{dt} = x_1(t)(\gamma - x_3(t)) - x_2(t), \tag{76}$$

$$\frac{dx_3(t)}{dt} = x_1(t)x_2(t) - \beta x_3(t), \tag{77}$$

where $\alpha = 10$, $\beta = 8/3$, and $\gamma = 28$. For this selection of parameters, the Lorenz system has chaotic solutions. Almost all initial points will tend to the invariant set, the Lorenz attractor. In the presented application, the initial condition is assumed to be Gaussian with mean $\hat{\mathbf{x}} = \begin{bmatrix}10 & 10 & 10\end{bmatrix}^T$ and diagonal covariance matrix, with the same standard deviation for each component of the state: $\sigma_{\mathbf{x}} = 2.5$. The state is integrated in time at 30 Hz, with observations taken at each time-step. The measurement model consists of the range of the state from origin

$$y_k = \sqrt{x_1(t_k)^2 + x_2(t_k)^2 + x_3(t_k)^2} + \mu_k, \tag{78}$$

where measurement noise is assumed to be Gaussian with zero mean and standard deviation $\sigma_\mu = 1$.

Fig. 6 shows, on the top, one of the trajectories described by the state of the system, in its three components. The Lorenz attractor has two main lobes symmetric with respect to the x_3 axis: The resulting pathway has been labeled a "butterfly" shape. A Monte Carlo analysis with 1000 realizations with SACEMM-2-5-2 is reported at the bottom of Fig. 6. For each ith component of the state, the estimation error of each realization is calculated according to Equation (73), and reported in gray. Analogously with the previous application, the continuous blue lines represent the predicted error standard deviations, as 3σ values, of each component, while the dashed blue lines are the effective error standard deviations, again as 3σ values, calculated directly from the Monte Carlo realizations at each time-step. The overlapping between the dashed and the continuous lines indicates that SACEMM-2-3-2 is a consistent filter able to correctly estimate its own uncertainties. The black lines are the error means, and they prove the unbiased nature of the proposed filtering technique, as expected from the MMSE theory.

The performance of the filters have been assessed through a covariance comparison carried out with multiple Monte Carlo analyses, each performed with 1000

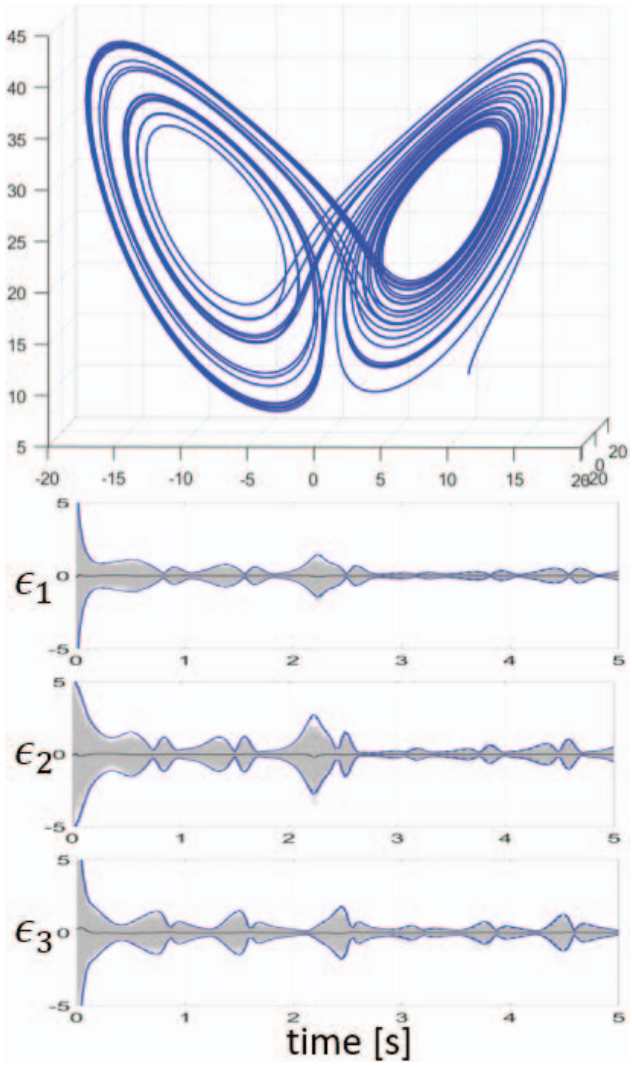


Fig. 6. Trajectory and SACEMM-2-3-2 Monte Carlo analysis results: 1000 runs.

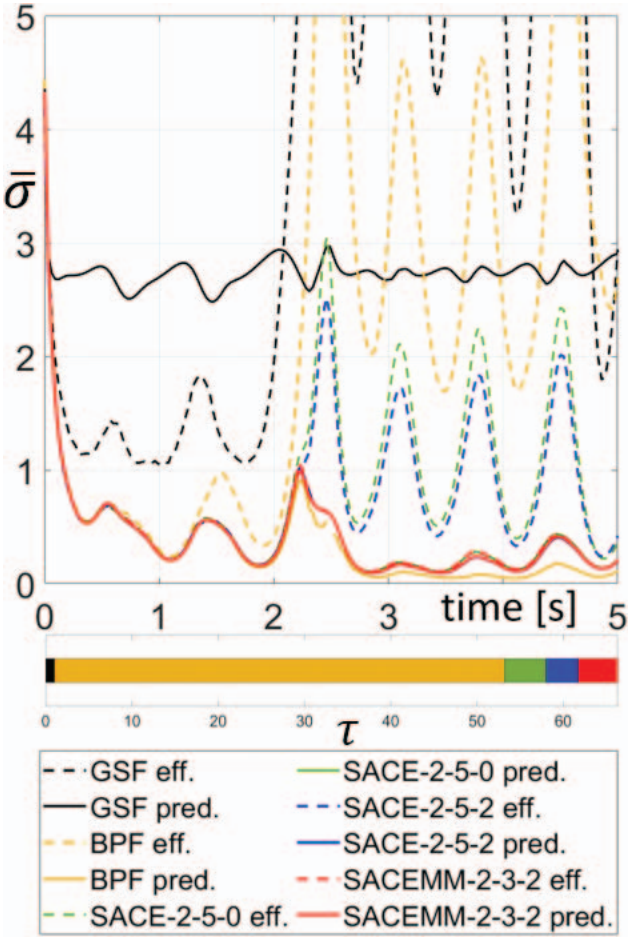


Fig. 7. Lorenz63: covariance comparison and time analysis among different filters.

runs. Fig. 7 reports, for each filter, the effective and the predicted error standard deviations. As shown in previous analysis, the dashed lines represent the actual uncertainty level of the filter, while the continuous lines are the filter's own uncertainty estimate, evaluated according to Equation (74). Fig. 7 reports SACE-c-η-μ and SACEMM-c-η-μ with different sets of parameters. For the basic selection of SACEMM-1-1-0, the filter reduces to the GSF, where the dynamics are linearized around the current center of each model, and the update is a linear estimator. The GSF is reported with black lines, and it fails to estimate the state of the system. The effective covariance indicates divergence and goes out of scale with respect to the predicted standard deviation.

The state of the system is also estimated with a 10^4 particles BPF, shown in orange, and the IEKF. The IEKF diverges rapidly and is not reported in the figure since the errors quickly reach out-of-scale large values. The linearization of the dynamics employed by the IEKF is not sufficient to correctly propagate the state covariance forward in time. The divergence of the IEKF might be connected to the poor time propagation. However, this issue might be alleviated by using the Levenberg–Marquardt algorithm [39]. The BPF performs better than the GSF but shows convergency problems, and it is not able to achieve an accurate estimate of the state. The BPF has issues due to the lack of process noise in the dynamics. After resampling, the propagated particles are not spread enough to be an appropriate representation of the prior uncertainty in order to accurately perform the measurement update.

SACE-c-η-μ is analyzed with the traditional zeroth-order covariance estimation, SACE-2-5-0 shown in green, and with a second-order covariance polynomial estimator, SACE-2-5-2 in blue. The two filters behave similarly: They both show convergence with consistency for the first half of the simulation, and they diverge for the remaining half. At time-step $t = 2.2$ s, the state of the system is near the origin, in between the two lobes of the Lorenz attractor. This point is critical because, due to uncertainty, the estimated state may select the incorrect lobe, while the true state follows the other. The measurement model, consisting solely of the range, gives no beneficial information regarding the lobe selection: Thus, the correction terms in the update step

do not help tracking the state along the correct path. Consequently, in some realizations of the Monte Carlo analysis, the filter is tracking the state of the system as if it were on the incorrect lobe. The radial nature of the range measurement provides no information to the estimator about correcting the estimated state because of the symmetric nature of the "butterfly" trajectory. Therefore, both SACE-2-5-0 and SACE-2-5-2 show inconsistency after the critical point, and the effective standard deviation is bigger than the predicted one. However, it is worth noticing that the dashed blue line settles below the dashed green line, indicating an increase in accuracy achieved due to the estimated covariance being connected with the measurement outcome. Lastly, SACEMM-2-3-2 is reported in red, and it is the only filter that shows convergency and consistency during the whole length of the simulation. The introduction of multiple models improves accuracy, especially around the critical point, where smaller subdomains make it easier for the filter to follow the right path along the correct lobe. If a model separates from the others, following the incorrect lobe, then it is weighted down in order to ensure a correct estimation. The division of the system uncertainties in smaller subdomains helps the filter track the correct trajectory, while the high-order polynomial update ensures excellent accuracy levels. SACEMM-c-η-μ has better performance than SACE-c-η-μ when the initial uncertainties of the state of the system are exceptionally high and when the propagated state PDF is multimodal.

The second part of Fig. 7 reports an analysis on the computational time requested by each filter. The parameter τ is evaluated as

$$\tau = \frac{\mathcal{T}_i}{\mathcal{T}_{\text{GSF}}}, \tag{79}$$

where $\mathcal{T}_i$ is the computational time of the ith filter, with $i = \{\text{GSF, BPF, SACE-2-5-0, SACE-2-5-2, SACEMM-2-3-2}\}$. Therefore, the τ bar expresses the relative computational effort among the different filters for this application. The τ analysis shows that the BPF is the computationally heaviest filter, while the computational time requested by SACE-c-η-μ and SACEMM-c-η-μ changes depending on the selection of their parameters.

VI. CONCLUSIONS

A novel filter based on a double estimator has been presented. The new technique estimates the conditional mean and the conditional covariance of the posterior distribution by applying, sequentially, two polynomial estimators, using the same measurement outcome. The new approach better matches the estimated state with its error standard deviation, which is now a polynomial function of the measurement. Therefore, the newly proposed filter is able to reduce the error uncertainty when the posterior distribution gets narrower around a low probability realization of the measurement. In turn, the better representation of the uncertainty produces a better estimate of the state during the subsequent measurement updates.

Three numerical examples have been reported. The scalar application gives a visual representation of the benefits of the polynomial approximation of the true MMSE and its covariance. Thus, the higher the order of the updates, the more precise the relative state estimate and its covariance. The vectorial application underlines the benefits of predicting the covariance by considering its estimation as working with an augmented state. The new state estimate improves in accuracy and a smaller error standard deviation is obtained. The multiple-model filter is more robust against high initial standard deviations and multimodal distributions.

REFERENCES

[1] D. Alspach and H. Sorenson
"Nonlinear Bayesian estimation using Gaussian sum approximations,"
IEEE Trans. Autom. Control, vol. 17, no. 4, pp. 439–448, Aug. 1972.

[2] I. Arasaratnam and S. Haykin
"Cubature Kalman filters,"
IEEE Trans. Autom. Control, vol. 54, no. 6, pp. 1254–1269, Jun. 2009.

[3] I. Arasaratnam, S. Haykin, and R. J. Elliot
"Discrete-time nonlinear filtering algorithms using Hauss–Hermite quadrature,"
Proc. IEEE, vol. 95, no. 5, pp. 953–977, May 2007.

[4] R. Armellin and P. Di Lizia
"Probabilistic optical and radar initial orbit determination,"
J. Guid., Control, Dyn., vol. 41, no. 1, pp. 101–108, 2018.

[5] R. Armellin, P. Di Lizia, F. Bernelli-Zazzera, and M. Berz
"Asteroid close encounters characterization using differential algebra: The case of Apophis,"
Celestial Mech. Dynamical Astron., vol. 107, no. 4, pp. 451–470, 2010.

[6] B. Bell and F. W. Cathey
"The iterated Kalman filter update as a Gauss–Newton method,"
IEEE Trans. Autom. Control, vol. 38, no. 2, pp. 294–297, Feb. 1993.

[7] M. Berz
Modern Map Methods in Particle Beam Physics. New York, NY, USA: Academic Press, 1999.

[8] F. Carravetta, A. Germani, and N. Raimondi
"Polynomial filtering of discrete-time stochastic linear systems with multiplicative state noise,"
IEEE Trans. Autom. Control, vol. 42, no. 8, pp. 1106–1126, Aug. 1997.

[9] F. Cavenago, P. Di Lizia, M. Massari, and A. Wittig
"On-board spacecraft relative pose estimation with high-order extended Kalman filter,"
Acta Astronautica, vol. 158, pp. 55–67, 2019.

[10] F. Cavenago, P. Di Lizia, M. Massari, S. Servadio, and A. Wittig
"DA-based nonlinear filters for spacecraft relative state estimation,"
AIAA 2018-1964. *2018 Space Flight Mechanics Meeting*. January 2018. doi: 10.2514/6.2018-1964.

[11] A. De Santis, A. Germani, and M. Raimondi
"Optimal quadratic filtering of linear discrete-time non-Gaussian systems,"
IEEE Trans. Autom. Control, vol. 40, no. 7, pp. 1274–1278, Jul. 1995.

[12] P. Di Lizia, M. Massari, F. Cavenago, and A. Wittig
"Assessment of onboard DA state estimation for spacecraft relative navigation," ESA, Paris, France, Contract IPL-PTE/LF/as/517.2016, Final Rep., 2017.

[13] A. Gelb
Applied Optimal Estimation. Cambridge, MA, USA: MIT Press, 1974.

[14] J. R. Fisher
"Optimal nonlinear filtering,"
Advances in Control Systems, vol. 5, pp. 197–300, 1967.

[15] N. J. Gordon, D. J. Salmond, and A. FM. Smith
"Novel approach to nonlinear/non-Gaussian Bayesian state estimation,"
IEE Proc. F-Radar Signal Process., vol. 140, no. 2, pp. 107–113, 1993.

[16] F. Hutter and R. Dearden
"The Gaussian particle filter for diagnosis of non-linear systems,"
IFAC Proc. Volumes, vol. 36, no. 5, pp. 909–914, 2003.

[17] L. Isserlis
"On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables,"
Biometrika, vol. 12, no. 1/2, pp. 134–139, 1918.

[18] A. H. Jazwinski
Stochastic Processes and Filtering Theory (Mathematics in Sciences and Engineering 64). New York, NY, USA: Academic Press, 1970.

[19] S. J. Julier and J. K. Uhlmann
"Unscented filtering and nonlinear estimation,"
Proc. IEEE, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[20] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte
"A new method for the nonlinear transformation of means and covariances in filters and estimators,"
IEEE Trans. Autom. Control, vol. 45, no. 3, pp. 477–482, Mar. 2000.

[21] R. E. Kalman
"A new approach to linear filtering and prediction problems,"
J. Basic Eng., vol. 82, no. 1, pp. 35–45, 1960.

[22] R. E. Kalman and R. S. Bucy
"New results in linear filtering and prediction theory,"
J. Basic Eng., vol. 83, no. 1, pp. 95–108, 1961.

[23] J. Lan and X. R. Li
"Nonlinear estimation based on conversion-sample optimization,"
Automatica, vol. 121, pp. 1873–2836, 2020.

[24] J. Lan and X. R. Li
"Nonlinear estimation by LMMSE-based estimation with optimized uncorrelated augmentation,"
IEEE Trans. Signal Process., vol. 63, no. 16, pp. 4270–4283, Aug. 2015.

[25] T. Lefebvre, H. Bruyninckx, and J. De Schutter
"Comment on 'A new method for the nonlinear transformation of means and covariances in filters and estimators'"
IEEE Trans. Autom. Control, vol. 47. no. 8, pp. 1406–1408, 2002.

[26] Y. Liu, X. R. Li, and H. Chen
"Generalized linear minimum mean-square error estimation with application to space-object tracking,"
in *Proc. Asilomar Conf. Signals, Syst. and Comput.*, 2013, pp. 2133–2137.

[27] K. Makino and M. Berz
"Cosy infinity version 9, nuclear instruments and methods in physics research section A: Accelerators,"
Spectrometers, Detectors Associated Equip., vol. 558, no. 1, pp. 346–350, 2006.

[28] M. Massari, P. Di Lizia, F. Cavenago, and A. Wittig
"Differential Algebra software library with automatic code generation for space embedded applications," in *Proc. AIAA Inf. Syst.-AIAA Infotech@ Aerospace*, 2018, p. 0398.

[29] M. Murata, H. Nagano, and K. Kashino
"Monte Carlo filter particle filter,"
in *Proc. Eur. Control Conf.*, 2015, pp. 2836–2841.

[30] D. Raihan and S. Chakravorty
"Particle Gaussian mixture filters-I,"
Automatica, vol. 98, pp. 331–340, 2018.

[31] O. Ravn, M. Norgaard, and N. K. Poulsen
"New developments in state estimations for nonlinear systems,"
Automatica, vol. 36, no. 11, pp. 1627–1638, 2000.

[32] T. S. Schei
"A finite-difference method for linearization in nonlinear estimation algorithms,"
Automatica, vol. 33, no. 11, pp. 2053–2058, 1997.

[33] T. Schon, F. Gustafsson, and P.-J. Nordlund
"Marginalized particle filters for mixed linear/nonlinear state-space models,"
IEEE Trans. Signal Process., vol. 53, no. 7, pp. 2279–2289, Jul. 2005.

[34] S. Servadio
"High order filters for relative pose estimation of an uncooperative target," M.S. thesis, Politecnico di Milano, Milan, Italy, 2017.

[35] S. Servadio and R. Zanetti
"Differential algebra-based multiple Gaussian particle filter for orbit determination," *J. Optim. Theory Appl.* (2021). https://doi.org/10.1007/s10957-021-01934-8.

[36] S. Servadio and R. Zanetti
"Recursive polynomial minimum mean-square error estimation with applications to orbit determination,"
J. Guid., Control, Dyn., AIAA, vol. 45, no. 5, pp. 939–954, 2020.

[37] S. Servadio, R. Zanetti, and R. Armellin
"Maximum a posteriori estimation of hamiltonian systems with high order series expansions,"
in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference held August 11–15, 2019, Portland, Maine, U.S.A.*, Volume 171 of Advanced in the Astronautical Sciences, Univelt, San Diego. pp. 2843–2858.

[38] S. Servadio, R. Zanetti, and B. A. Jones
"Nonlinear filtering with a polynomial series of G aussian random variables,"
IEEE Trans. Aerosp. Electron. Syst., vol. 57, no. 1, pp. 647–658, Feb. 2021.

[39] M. A. Skoglund, G. Hendeby, and D. Axehill
"Extended Kalman filter modifications based on an optimization view point," in *Proc. 18th Int. Conf. Inf. Fusion*, 2015, pp. 1856–1861, 2015.

[40] H. W. Sorenson and D. L. Alspach
"Recursive Bayesian estimation using Gaussian sums,"
Automatica, vol. 7, no. 4, pp. 465–479, 1971.

[41] G. Terejanu, P. Single, T. Singh, and P. D. Scott
"Adaptive Gaussian sum filter for nonlinear Bayesian estimation,"
IEEE Trans. Autom. Control, vol. 56, no. 9, pp. 2151–2156, Sep. 2011.

[42] M. Valli, R. Armellin, P. Di Lizia, and M. Lavagna
"A Gaussian particle filter based on differential algebra for spacecraft navigation,"
in *Proc. 63rd Int. Astronautical Congr.*, International Astronautical Federation, 2012, pp. 5178–5187.

[43] M. Valli, R. Armellin, P. Di Lizia, and M. Lavagna
"Nonlinear mapping of uncertainties in celestial mechanics,"
J. Guid., Control, Dyn., vol. 36, no. 1, pp. 48–63, 2012.

[44] R. van der Merwe, A. Doucet, N. De Freitas, and E. A. Wan
"The unscented particle filter,"
in *Proc. Advances Neural Inf. Process. Syst.*, 2001, pp. 584–590.

[45] A. Wittig, P. Di Lizia, R. Armellin, K. Makino, F. Bernelli-Zazzera, and M. Berz
"Propagation of large uncertainty sets in orbital dynamics by automatic domain splitting,"
Celestial Mechanics Dynamical Astron., vol. 122, no. 3, pp. 239–261, 2015.

[46] Y. Zhang and J. Lan
"Gaussian sum filtering using uncorrelated conversion for nonlinear estimation,"
in *Proc. 20th Int. Conf. Inf. Fusion*, 2017, pp. 1–8.

Simone Servadio is a Postdoctoral Associate with the Massachusetts Institute of Technology, Cambridge, MA, USA. He received the Doctorate degree from the University of Texas at Austin, Austin, TX, USA, studying nonlinear estimation, orbit determination, attitude estimation and control, tracking, and GNC applications. His current research interests include astrodynamics and the Koopman operator, with particular interest for stable and unstable manifolds, periodic orbits, and the circular restricted three-body problem.

Renato Zanetti is an Assistant Professor with the Aerospace Engineering, University of Texas at Austin, Austin, TX, USA. Prior to joining the University of Texas at Austin, he worked as an Engineer with the NASA Johnson Space Center and with Draper Laboratory. His research interests include nonlinear estimation, onboard navigation, and autonomous aerospace vehicles.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

ISIF Website: http://www.isif.org

2021 BOARD OF DIRECTORS*

2019–2021	2020–2022	2021–2023
Kathryn Laskey	Pieter De Villiers	Alta De Waal
Felix Govaers	Murat Efe	Fredrik Gustafsson
Simon Maskell	Wolfgang Koch	Uwe Hanebeck

*Board of Directors are elected by the members of ISIF for a three year term.

PAST PRESIDENTS

Paulo Costa, 2020	Roy Streit, 2012	Chee Chong, 2004
Paulo Costa, 2019	Joachim Biermann, 2011	Xiao-Rong Li, 2003
Lyudmila Mihaylova, 2018	Stefano Coraluppi, 2010	Yaakov Bar-Shalom, 2002
Lyudmila Mihaylova, 2017	Elisa Shahbazian, 2009	Pramod Varshney, 2001
Jean Dezert, 2016	Darko Musicki, 2008	Yaakov Bar-Shalom, 2000
Darin Dunham, 2015	Erik Blasch, 2007	Jim Llinas, 1999
Darin Dunham, 2014	Pierre Valin, 2006	Jim Llinas, 1998
Wolfgang Koch, 2013	W. Dale Blair, 2005	

SOCIETY VISION

The International Society of Information Fusion (ISIF) is the premier professional society and global information resource for multidisciplinary approaches for theoretical and applied information fusion technologies.

SOCIETY MISSION

Advocate

To advance the profession of fusion technologies, propose approaches for solving real-world problems, recognize emerging technologies, and foster the transfer of information.

Serve

To serve its members and engineering, business, and scientific communities by providing high-quality information, educational products, and services.

Communicate

To create international communication forums and hold international conferences in countries that provide for interaction of members of fusion communities with each other, with those in other disciplines, and with those in industry and academia.

Educate

To promote undergraduate and graduate education related to information fusion technologies at universities around the world. Sponsor educational courses and tutorials at conferences.

Integrate

Integrate ideas from various approaches for information fusion, and look for common threads and themes– look for overall principles, rather than a multitude of point solutions. Serve as the central focus for coordinating the activities of world-wide information fusion related societies or organizations. Serve as a professional liaison to industry, academia, and government.

Disseminate

To propagate the ideas for integrated approaches to information fusion so that others can build on them in both industry and academia.

Call for Papers

The Journal of Advances in Information Fusion (JAIF) seeks original contributions in the technical areas of research related to information fusion. Authors are encouraged to submit their manuscripts for peer review http://isif.org/journal.

Call for Reviewers

The success of JAIF and its value to the research community is strongly dependent on the quality of its peer review process. Researchers in the technical areas related to information fusion are encouraged to register as a reviewer for JAIF at http://jaif.msubmit.net. Potential reviewers should notify via email the appropriate editors of their offer to serve as a reviewer.

Journal of Advances in Information Fusion Volume 16 Number 2 December 2021 ISSN 1557-6418