

Journal of Advances in Information Fusion

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

Regular Papers	Page
Bearings-Only Tracking with Fusion from Heterogenous Passive Sensors: ESM/EO and Acoustic	3
<i>Rong Yang, DSO National Laboratories, Singapore</i>	
<i>Yaakov Bar-Shalom, University of Connecticut, USA</i>	
<i>Gee Wah Ng, DSO National Laboratories, Singapore</i>	
Joint Identification of Multiple Tracked Targets	20
<i>Dominic E. Schaub, Atlantic Research Centre, Canada</i>	
Cognitive Video Streaming	41
<i>D. Pasupuleti, University of Connecticut, USA, University of Connecticut, USA</i>	
<i>P. Mannaru, University of Connecticut, USA</i>	
<i>B. Balasingam, University of Connecticut, USA</i>	
<i>M. Baum, University of Connecticut, USA</i>	
<i>K. Pattipati, University of Connecticut, USA</i>	
<i>P. Willett, University of Connecticut, USA</i>	
<i>C. Lintz, Comcast Corporation, USA</i>	
<i>G. Commeau, Comcast Corporation, USA</i>	
<i>F. Dorigo, Comcast Corporation, USA</i>	
<i>J. Fahrny, Comcast Corporation, USA</i>	
Space Based Sensor Bias Estimation in the Presence of Data Association Uncertainty	58
<i>Djedjiga Belfadel, University of Connecticut, USA</i>	
<i>Richard W. Osborne, III, University of Connecticut, USA</i>	
<i>Yaakov Bar-Shalom, University of Connecticut, USA</i>	
<i>Krishna Pattipati, University of Connecticut, USA</i>	
Analysis of Log-Homotopy Based Particle Flow Filters	73
<i>Muhammad Altamash Khan, Fraunhofer FKIE, Wachtberg, Germany</i>	
<i>Martin Ulmke, Fraunhofer FKIE, Wachtberg, Germany</i>	
<i>Wolfgang Koch, Fraunhofer FKIE, Wachtberg, Germany</i>	
Applying Interacting Multiple Model to Financial Asset Allocation	95
<i>Shozo Mori, Systems & Technology Research, USA</i>	
<i>K. C. Chang, George Mason University, USA</i>	
<i>Hajime Takahashi, Tottori University of Environmental Studies, Japan</i>	
<i>Chee-Yee Chong, Independent Researcher, USA</i>	
Tracking a Maneuvering Target Using Two Heterogeneous Passive Sensors on a Single Stationary Platform with IMM Estimation	110
<i>Hong An Jack Huang, DSO National Laboratories, Singapore</i>	
<i>Yaakov Bar-Shalom, University of Connecticut, USA</i>	
<i>Rong Yang, DSO National Laboratories, Singapore</i>	
<i>Gee Wah Ng, DSO National Laboratories, Singapore</i>	
Volumes 1-11 Index	125

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: June 2017

Editor-In-Chief	Uwe D. Hanebeck	Karlsruhe Institute of Technology (KIT), Germany; +49-721-608-43909; uwe.hanebeck@ieee.org
Associate	Stefano Coraluppi	Systems & Technology Research, USA; +1 781-305-4055; stefano.coraluppi@ieee.org
Administrative Editor	David W. Krout	University of Washington, USA; +1 206-616-2589; dkrou@apl.washington.edu
Associate	Ruixin Niu	Virginia Commonwealth University, Richmond, Virginia, USA; +1 804-828-0030; rniu@vcu.edu
Associate	Marcus Baum	Karlsruhe Institute of Technology (KIT), Germany; +49-721-608-46797; marcus.baum@kit.edu

EDITORS FOR TECHNICAL AREAS

Tracking	Stefano Coraluppi	Systems & Technology Research, USA; +1 781-305-4055; stefano.coraluppi@ieee.org
Associate	Paolo Braca	NATO Science & Technology Organization, Centre for Maritime Research and Experimentation, Italy; +39 0187 527 461; paolo.braca@cmre.nato.int
Detection	Pramod Varshney	Syracuse University, Syracuse, New York, USA; +1 315-443-1060; varshney@syr.edu
Fusion Applications	Ben Slocumb	Numerica Corporation; Loveland, Colorado, USA; +1 970-461-2000; bjslocumb@numerica.us
Associate	Ramona Georgescu	United Technologies Research Center, East Hartford, Connecticut, USA; 860-610-7890; georgera@utrc.utc.com
Image Fusion	Lex Toet	TNO, Soesterberg, 3769de, Netherlands; +31 346356237; lex.toet@tno.nl
Associate	Ting Yuan	Mercedes Benz R&D North America, USA; +1 669-224-0443; dr.ting.yuan@ieee.org
High-Level Fusion	Lauro Snidaro	Università degli Studi di Udine, Udine
Fusion Architectures and Management Issues	Chee Chong	BAE Systems, Los Altos, California, USA; +1 650-210-8822; chee.chong@baesystems.com
Classification, Learning, Bayesian and Other Reasoning Methods	Nageswara S. V. Rao	Oak Ridge National Laboratory, USA; +1 865-574-7517;
Associate	Claude Jauffret	Université de Toulon, La Garde, France; +33 (0) 4 94 14 24 14; jauffret@univ-tln.fr
	Jean Dezert	ONERA, Chatillon, 92320, France; +33 146734990; jean.dezert@onera.fr

Manuscripts are submitted at <http://jaif.msubmit.net>. If in doubt about the proper editorial area of a contribution, submit it under the unknown area.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

Mila Mihaylova, *President*

Lyudmila Mihaylova, *President-elect*

Stefano Coraluppi, *Secretary*

Chee Chong, *Treasurer*

Dale Blair, *Vice President Publications*

David W. Krout, *Vice President Communications*

Lance Kaplan, *Vice President Conferences*

Anne-Laure Jousselme, *Vice President Membership*

Paulo Costa, *Vice President Working Groups*

Uwe Hanebeck, *JAIF EIC*

Roy Streit, *Perspectives EIC*

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. **Copyright and Reprint Permissions:** Abstracting is permitted with credit to the source. For all other copying, reprint, or republication permissions, contact the Administrative Editor. Copyright© 2017 ISIF, Inc.

Bearings-Only Tracking with Fusion from Heterogenous Passive Sensors: ESM/EO and Acoustic

RONG YANG
YAAKOV BAR-SHALOM
GEE WAH NG

The performance of the conventional bearings-only tracking (BOT) from a single passive sensor hinges on the sensor platform maneuvers. This paper presents a new BOT approach based on fusion from two heterogenous bearings-only sensors residing on the same moving or stationary platform. The two sensors are an ESM/EO with negligible propagation delay and an acoustic sensor with significant propagation delay. The time difference between the reception times of the two sensors (corresponding to the same emission time) is the acoustic propagation delay. Since target range information is contained in the acoustic propagation delay (which is not known but can be estimated), the target state is shown to be completely observable even when the platform is stationary. The observability is studied in this paper via the Fisher information matrix (FIM).

Two estimators are developed. They are the maximum likelihood (ML) estimator for batch estimation and the out-of-sequence measurements fusion from acoustic and ESM/EO sensors (OOSM-AE) for recursive estimation. It shows that the ML estimator for batch estimation attains the Cramér-Rao lower bound (CRLB)—it is statistically efficient—except in cases with a small number of measurements and the target heading close to the bearing from the sensor platform. The OOSM-AE is developed to handle out-of-sequence measurements (OOSM) due to the acoustic propagation delay. It consists of an unscented Kalman filter (UKF) to handle the in-sequence ESM/EO measurements and an OOSM unscented Gauss-Helmert filter (OOSM-UGHF) to handle the out-of-sequence acoustic measurements. Simulation results are presented to demonstrate the performance of this new BOT approach.

Manuscript received April 19, 2015; revised November 30, 2015 and April 6, 2016; released for publication May 4, 2016.

Refereeing of this contribution was handled by Paolo Braca.

Authors' addresses: R. Yang and G. Wah Ng, DSO National Laboratories, 20 Science Park Drive, Singapore 118230 (E-mail: ryoung@dso.org.sg; ngeewah@dso.org.sg); Y. Bar-Shalom, Department of ECE, University of Connecticut, Storrs, CT 06269, USA (E-mail: ybs@engr.uconn.edu).

Y. Bar-Shalom is supported by ARO Grant W911NF-10-1-0369.

1557-6418/17/\$17.00 © 2017 JAIF

I. INTRODUCTION

The commonly used passive sensors, like acoustic sensors, electronic support measures (ESM) sensors and electro-optical (EO) sensors, measure target bearings only. This makes the target state estimation from range-absent measurements a challenging problem.

Several approaches for this problem have been developed in the last four decades. The most popular one is to deploy a passive sensor on a maneuvering platform, and the target state is estimated using bearings-only tracking (BOT) or bearings-only target motion analysis (BO-TMA) [16] [1]. This approach requires the sensor platform to maneuver, so the target state is observable [17] [11] [6]. Since these maneuvers can interfere with the sensor platform's own mission (for example: to reach its destination as early as possible), BOT from a nonmaneuvering platform has attracted attention recently. Results showed that the BOT problem is indeed observable from a nonmaneuvering platform when the target is performing particular maneuvers (two-leg with constant speed, or constant turn) [13] [7]. However, there is still a gap to transition these results to real applications, for the target can maneuver in a manner unbeknownst to the observer.

The BOT approach has been extended to the Doppler-bearing tracking (DBT) approach in [18] [10]. This approach tracks the target state and emitted frequencies from bearing and Doppler shifted frequency measurements and the state can be estimated even when the platform is not maneuvering. The difficulty faced in DBT is to identify the target Doppler shifted frequencies from a noisy environment, especially when the target emitted frequencies are varying.

Another approach is to locate targets through triangulation from multiple stationary or moving passive sensors located at different positions. This approach needs to remove triangulation “ghosts” in multi-target scenarios, and can be solved as an S -D assignment problem, where S is the number of sensors. A Lagrangian relaxation approach was suggested to solve this problem when $S \geq 3$ [19] [8]. By making use of Doppler frequencies, the number of sensors can be reduced to 2 ($S = 2$) [22].

In this paper, we propose a new bearings-only approach to fuse measurements from two heterogenous passive sensors deployed on the same platform which can be *either moving or stationary*. The two sensors are a passive ESM/EO sensor, designated as s_1 and a passive acoustic sensor, designated as s_2 . Both sensors measure target bearings only. The ESM/EO sensor's detections have no propagation delay, whereas the acoustic sensor receives the target signals after significant propagation delays. The time difference between the reception times of the two sensors (corresponding to the same emission time) is the acoustic propagation delay, and the target range can then be inferred from the estimates of

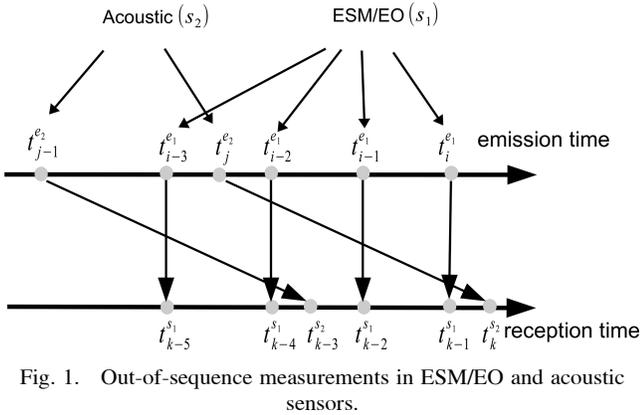


Fig. 1. Out-of-sequence measurements in ESM/EO and acoustic sensors.

these delays assuming the propagation speed is known. Complete observability in this BOT problem is therefore obtained, as range is implied in the sensors' reception times.

However, to obtain target range using the principle mentioned above is not straightforward. To compute the acoustic propagation delay, a pair of passive signals from s_1 and s_2 having the same emission time needs to be identified. A BOT target usually emits continuous signals which are received by the sensors and discretized by sampling. They are not instantaneous signals, like "ping" or "pulse" which can be associated easily. There is no feature to identify an acoustic bearing measurement and an ESM/EO measurement emitted at the same time. Furthermore, the ESM/EO and acoustic sensor may have different sampling times (they are asynchronous), and the sensor platform may be dynamic. These make the problem even more complicated.

Fig. 1 illustrates the ESM/EO and acoustic signal emission and reception time sequences, where k is the reception time index, which orders the combined acoustic and EO/ESM discretized signals by arrival (sensor) time. This is also the measurement index, while i and j are the target signal emission time indexes from s_1 and s_2 , respectively. It can be seen that measurements arrive the observer out-of-sequence due to the acoustic propagation delay.

Our preliminary study on this problem has been presented in [25] recently. At the same time, the problem has also been addressed in [14]. In the present paper, we will conduct a comprehensive study, which includes the problem observability, appropriate algorithms from batch and recursive estimation and analyzing the efficiency of these estimators.

The structure of the rest of paper is as follows. Section II formulates the problem as a batch estimation problem, and develops a maximum likelihood (ML) estimator for the problem. Section III analyzes the observability of the problem via the Fisher Information Matrix (FIM). Section IV presents simulation results

for the batch estimation and studies the estimator efficiency. Sections V–VII focus on the recursive estimation. Sections V and VI develop a novel recursive state estimation algorithm, and Section VII presents simulation results and recursive estimator efficiency analysis. Conclusions are given in Section VIII.

II. PROBLEM FORMULATION AND ML ESTIMATOR FOR THE BATCH ESTIMATION

This section formulates the acoustic and ESM/EO bearing fusion problem (designated as "AE") as a parameter estimation problem from a batch of bearing measurements. Since the BOT problem has been well studied when the platform is maneuvering, we focus on the stationary platform here. The target motion parameter \mathbf{x} is to be estimated from the measurement vector \mathbf{Z} consisting of a batch of ESM/EO (s_1) and acoustic (s_2) bearings. This is modeled as

$$\mathbf{Z} = \mathbf{h}(\mathbf{x}) + \mathbf{w} \quad (1)$$

where $\mathbf{h}(\cdot)$ is the function that relates \mathbf{x} to \mathbf{Z} , and \mathbf{w} is the measurement noise. The measurement vector \mathbf{Z} is

$$\mathbf{Z} = [b(t_1^s) \cdots b(t_n^s)]' \quad (2)$$

where $s \in \{s_1, s_2\}$ is the sensor receiving the signal at time t_k^s . The parameter \mathbf{x} consists of the position and velocity of the target at time t_n^s

$$\mathbf{x} = [x \ y \ \dot{x} \ \dot{y}]' \quad (3)$$

Assuming the measurement noises of s_1 and s_2 are zero-mean white Gaussian with the same standard deviation¹ σ_b , the covariance of \mathbf{w} is

$$\mathbf{R} = \sigma_b^2 \mathbf{I}_n \quad (4)$$

where \mathbf{I}_n is the identity matrix of dimension n .

The function that relates \mathbf{x} to \mathbf{Z} is

$$\mathbf{h}[\mathbf{x}] = [h(t_1^s, \mathbf{x}) \cdots h(t_n^s, \mathbf{x})]' \quad (5)$$

where $h(\cdot)$ is the function that maps \mathbf{x} to s_1 or s_2 bearings. Assuming the stationary sensor is located at $(0,0)$, $h(\cdot)$ is given by

$$h(t_k^s, \mathbf{x}) = \begin{cases} \tan^{-1} \left[\frac{x - (t_n^s - t_k^s)\dot{x}}{y - (t_n^s - t_k^s)\dot{y}} \right] & \text{if } s = s_1 \\ \tan^{-1} \left[\frac{x - (t_n^s - t_k^s + \delta_{j,k})\dot{x}}{y - (t_n^s - t_k^s + \delta_{j,k})\dot{y}} \right] & \text{if } s = s_2 \end{cases} \quad (6)$$

where $\delta_{j,k}$ is the propagation delay of the j th acoustic bearing with arrival time t_k^s (see (44)–(45) and Fig. 1), which follows from the quadratic equation

$$[x - (t_n^s - t_k^s + \delta_{j,k})\dot{x}]^2 + [y - (t_n^s - t_k^s + \delta_{j,k})\dot{y}]^2 = (c^p \delta_{j,k})^2 \quad (7)$$

¹This is for simplicity of notation only.

The solution of the above is²

$$\begin{aligned}\delta_{j,k} &= \frac{x_k^d \dot{x} + y_k^d \dot{y} \pm \phi_k}{\rho} \\ &= \frac{x_k^d \dot{x} + y_k^d \dot{y} - \phi_k}{\rho}\end{aligned}\quad (8)$$

where

$$\phi_k = \sqrt{(x_k^d \dot{x} + y_k^d \dot{y})^2 - [(x_k^d)^2 + (y_k^d)^2] \rho} \quad (9)$$

$$\rho = \dot{x}^2 + \dot{y}^2 - (c^p)^2 \quad (10)$$

$$x_k^d = x - (t_n^s - t_k^s) \dot{x} \quad (11)$$

$$y_k^d = y - (t_n^s - t_k^s) \dot{y} \quad (12)$$

The ML estimate $\hat{\mathbf{x}}$ of \mathbf{x} is obtained from the likelihood function $\Lambda(\mathbf{x}; \mathbf{Z})$ of \mathbf{x} based on the batch of measurements \mathbf{Z} as

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} \Lambda(\mathbf{x}; \mathbf{Z}) = \arg \max_{\mathbf{x}} p(\mathbf{Z} | \mathbf{x}) \quad (13)$$

Under the zero-mean Gaussian assumption on the noise \mathbf{w} , the above becomes the following nonlinear least squares (NLS) problem [2]

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \{[\mathbf{Z} - \mathbf{h}(\mathbf{x})]' \mathbf{R}^{-1} [\mathbf{Z} - \mathbf{h}(\mathbf{x})]\} \quad (14)$$

which will be solved numerically via the iterated squares (ILS) method. The ILS yields the ML estimate of the parameter $\mathbf{x}(n)$ (at the end of the batch of length n) is as follows

$$\mathbf{P}^l = [\mathbf{H}[\hat{\mathbf{x}}^l(n)]' \mathbf{R}^{-1} \mathbf{H}[\hat{\mathbf{x}}^l(n)]]^{-1} \quad (15)$$

$$\hat{\mathbf{x}}^{l+1}(n) = \hat{\mathbf{x}}^l(n) + \mathbf{P}^l \mathbf{H}[\hat{\mathbf{x}}^l(n)]' \mathbf{R}^{-1} [\mathbf{Z} - \mathbf{h}[\hat{\mathbf{x}}^l(n)]] \quad (16)$$

where l is the iteration number, and $\mathbf{H} = (\cdot)$ the Jacobian matrix of $\mathbf{h}(\cdot)$. This is derived next.

$$\mathbf{H}(\mathbf{x}) = (\nabla_{\mathbf{x}} \mathbf{h}[\mathbf{x}]')' = [\mathbf{H}_1^s \cdots \mathbf{H}_n^s]' \quad (17)$$

where

$$\mathbf{H}_k^s = \begin{cases} \left[\begin{array}{cccc} \frac{y_k^d}{(r_k^d)^2} & -\frac{x_k^d}{(r_k^d)^2} & -\frac{(t_n^s - t_k^s)y_k^d}{(r_k^d)^2} & \frac{(t_n^s - t_k^s)x_k^d}{(r_k^d)^2} \end{array} \right]' & \text{if } s = s_1 \\ [H_{k,1}^s \ H_{k,2}^s \ H_{k,3}^s \ H_{k,4}^s]' & \text{if } s = s_2 \end{cases} \quad (18)$$

and (recall that j denotes the index of the acoustic

²The negative sign is selected in (8.) so that the propagation delay $\delta_{j,k}$ is greater than 0, to match (6)–(7).

bearing that arrives at t_k)

$$H_{k,1}^s = \frac{y_{j,k}^e + (x_{j,k}^e \dot{y} - y_{j,k}^e \dot{x}) \nabla_x \delta_{j,k}}{(r_{j,k}^e)^2} \quad (19)$$

$$H_{k,2}^s = \frac{-x_{j,k}^e + (x_{j,k}^e \dot{y} - y_{j,k}^e \dot{x}) \nabla_y \delta_{j,k}}{(r_{j,k}^e)^2} \quad (20)$$

$$H_{k,3}^s = \frac{-(t_n^s - t_k^s + \delta_{j,k})y_{j,k}^e + (x_{j,k}^e \dot{y} - y_{j,k}^e \dot{x}) \nabla_x \delta_{j,k}}{(r_{j,k}^e)^2} \quad (21)$$

$$H_{k,4}^s = \frac{(t_n^s - t_k^s + \delta_{j,k})x_{j,k}^e + (x_{j,k}^e \dot{y} - y_{j,k}^e \dot{x}) \nabla_y \delta_{j,k}}{(r_{j,k}^e)^2} \quad (22)$$

$$\nabla_x \delta_{j,k} = \frac{1}{\rho} \left[\dot{x} - \frac{(c^p)^2 x_k^d - \dot{y}(x_k^d \dot{y} - y_k^d \dot{x})}{\phi_k} \right] \quad (23)$$

$$\nabla_y \delta_{j,k} = \frac{1}{\rho} \left[\dot{y} - \frac{(c^p)^2 y_k^d + \dot{x}(x_k^d \dot{y} - y_k^d \dot{x})}{\phi_k} \right] \quad (24)$$

$$\begin{aligned} \nabla_x \delta_{j,k} &= \frac{1}{\rho} \left[x_k^d - (t_n^s - t_k^s) \dot{x} + \frac{(c^p)^2 (t_n^s - t_k^s) x_k^d}{\phi_k} \right. \\ &\quad \left. - \frac{(x_k^d \dot{y} - y_k^d \dot{x})(y_k^d + (t_n^s - t_k^s) \dot{y})}{\phi_k} \right] \\ &\quad - \frac{2\dot{x}(x_k^d \dot{x} + y_k^d \dot{y} - \phi_k)}{\rho^2} \end{aligned} \quad (25)$$

$$\begin{aligned} \nabla_y \delta_{j,k} &= \frac{1}{\rho} \left[x_k^d - (t_n^s - t_k^s) \dot{x} + \frac{(c^p)^2 (t_n^s - t_k^s) y_k^d}{\phi_k} \right. \\ &\quad \left. + \frac{(x_k^d \dot{y} - y_k^d \dot{x})(x_k^d + (t_n^s - t_k^s) \dot{x})}{\phi_k} \right] \\ &\quad - \frac{2\dot{y}(x_k^d \dot{x} + y_k^d \dot{y} - \phi_k)}{\rho^2} \end{aligned} \quad (26)$$

$$r_k^d = \sqrt{(x_k^d)^2 + (y_k^d)^2} \quad (27)$$

$$x_{j,k}^e = x - (t_n^s - t_k^s + \delta_{j,k}) \dot{x} \quad (28)$$

$$y_{j,k}^e = y - (t_n^s - t_k^s + \delta_{j,k}) \dot{y} \quad (29)$$

$$r_{j,k}^e = \sqrt{(x_{j,k}^e)^2 + (y_{j,k}^e)^2} \quad (30)$$

III. OBSERVABILITY ANALYSIS VIA THE FISHER INFORMATION MATRIX

To analyze the observability of the nonlinear model (1), the Fisher information matrix (FIM) will be used. The relationship between the FIM and parameter observability (i.e., its estimability) has been studied in [12]. The parameter \mathbf{x} in (1) is completely observable, if the FIM is nonsingular (invertible). The FIM is given by [2]

$$\mathcal{F}(\mathbf{x}) = \mathbf{H}'(\mathbf{x}) \mathbf{R}^{-1} \mathbf{H}(\mathbf{x}) \quad (31)$$

where \mathbf{H} is the Jacobian matrix given in (17).

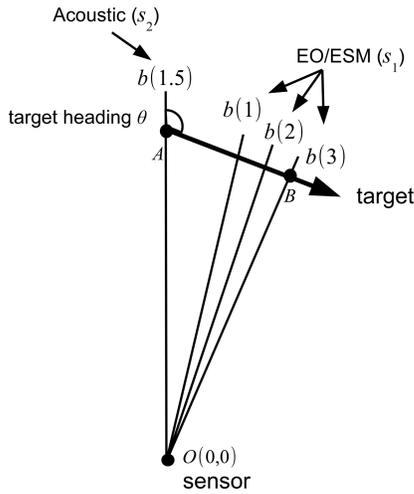


Fig. 2. Observability analysis. Estimation of the target position and velocity at point B using four bearings (the acoustic bearing's reception time is $t_2^2 = 1.5$ s).

Since the size of the parameter \mathbf{x} is four, \mathbf{Z} should consist of at least four bearings for observability. Fig. 2 illustrates the problem with three ESM/EO bearings received at times 1 s, 2 s and 3 s, and a delayed acoustic bearing with reception time 1.5 s, and

$$\mathbf{Z} = [b(1) \ b(1.5) \ b(2) \ b(3)]' \quad (32)$$

The parameter to be estimated is \mathbf{x} at time $t_4^1 = 3$ s, which corresponds to point B on the trajectory \overline{AB} .

The observability analysis is based on the numerical results of $\det[\mathcal{F}(\mathbf{x})]$ for various geometries of the target trajectory and platform. The scenarios consist of the target trajectory \overline{AB} rotating 360° around point A in Fig. 2, namely the target heading varies from 1° to 360° . The bearing error standard deviation for both sensors is $\sigma_b = 1^\circ$ in the FIM. Figs. 3–6 show $\det[\mathcal{F}(\mathbf{x})]$ versus target heading with the target speeds of 5 m/s, 10 m/s, 50 m/s and 100 m/s respectively. In each figure, $\det[\mathcal{F}(\mathbf{x})]$ is investigated at two different ranges, namely, target motion starting point A is at (0 m, 5000 m) and (0 m, 6000 m).

From the results, we observe that FIM is singular (or $\det[\mathcal{F}] = 0$) only when the target heading is at 180° and 360° w.r.t. the line-of-sight (LOS) to the sensor platform. The problem is unobservable in these cases, and the four bearings are the same and always in line with the target heading. Thus, we can conclude that the problem is completely observable unless the bearing is constant over time.

The FIM itself is the total information about the parameter \mathbf{x} from the measurement set \mathbf{Z} . A higher value of $\det[\mathcal{F}]$ represents a better estimation of \mathbf{x} from \mathbf{Z} . Obviously, the determinant of \mathcal{F} is affected by the target range, speed and heading. It can be seen that the value of $\det[\mathcal{F}]$ is increasing with target speed and decreasing with target range. For the target heading, it closely links to the change of the bearings in the batch. Intuitively, a larger bearing change gives a better

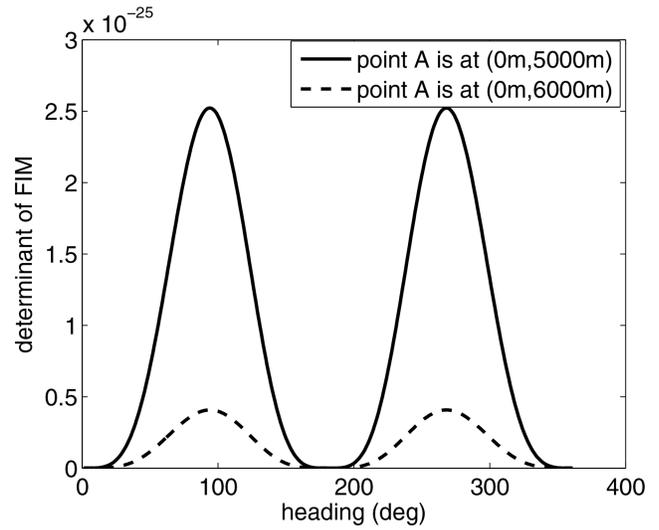


Fig. 3. $\det[\mathcal{F}]$ versus target heading when the target speed is 5 m/s.

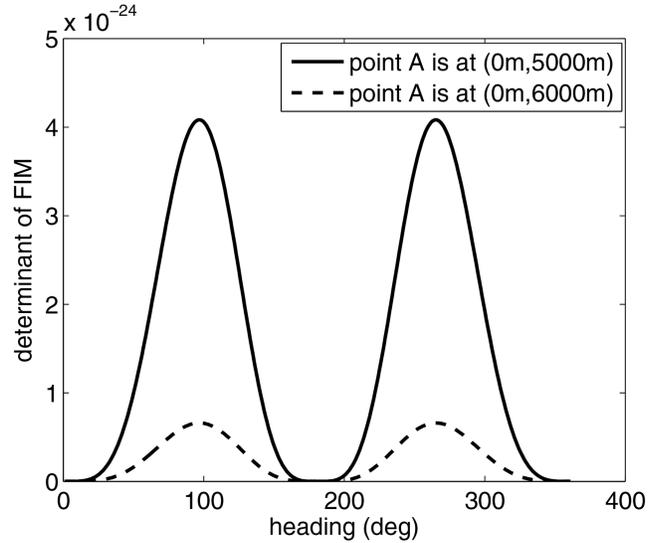


Fig. 4. $\det[\mathcal{F}]$ versus target heading when the target speed is 10 m/s.

estimate. We can see that $\det[\mathcal{F}]$ almost reaches zero when the heading is close to 180° or 360° , where the bearing change is small. When the heading is away from 180° and 360° , $\det[\mathcal{F}]$ increases as the bearing change increases. However, the maximum $\det[\mathcal{F}]$ is not exactly on the target trajectory with the largest change in bearing. Target range is also taken into consideration in FIM. Fig. 7 shows that the trajectory $\overline{AB'}$ has the largest change in bearing. However it does not have the highest $\det[\mathcal{F}]$. The trajectories with the highest $\det[\mathcal{F}]$ is \overline{AB} . This is because B is closer to the sensor than B' .

IV. SIMULATION RESULTS FOR THE BATCH ESTIMATION

The scenarios in the simulation are similar to those used in Section III but with different batch size. Point A in Fig. 7 is set at (0 m, 5000 m). The target speed is 100 m/s, and the heading is chosen from 10° – 140° .

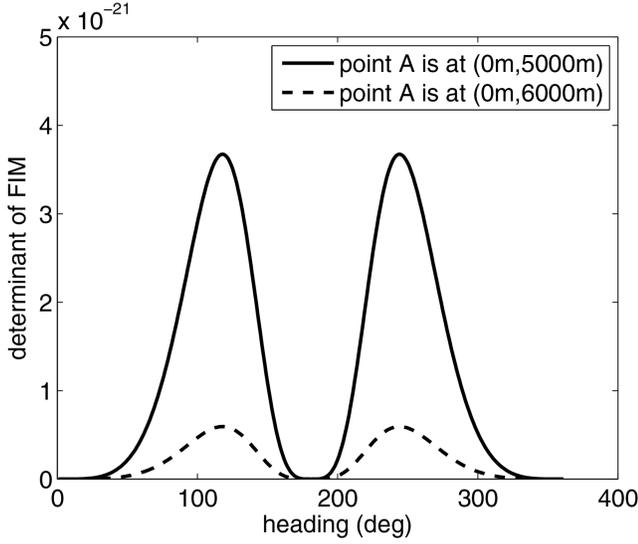


Fig. 5. $\det[\mathcal{F}]$ versus target heading when the target speed is 50 m/s.

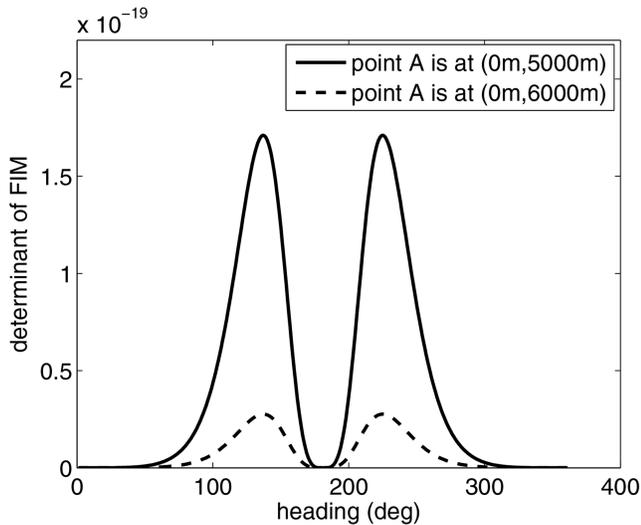


Fig. 6. $\det[\mathcal{F}]$ versus target heading when the target speed is 100 m/s.

The bearing error standard deviations for both s_1 and s_2 are set to $\sigma_b = 1^\circ$. Four different batch sizes are investigated, namely

- $n = 10$: 10 ESM/EO bearings and 5 acoustic bearings over a total time of 10 s.
- $n = 20$: 20 ESM/EO bearings and 10 acoustic bearings over a total time of 20 s.
- $n = 30$: 30 ESM/EO bearings and 15 acoustic bearings over a total time of 30 s.
- $n = 60$: 60 ESM/EO bearings and 30 acoustic bearings over a total time of 60 s.

This section also studies the CRLB of the problem, and compares the errors of the ILS estimates to the CRLB. The CRLB error covariance matrix \mathbf{P}^{CRLB} , which is a 4×4 matrix, is given by

$$\mathbf{P}^{\text{CRLB}} = \mathcal{F}^{-1} \quad (33)$$

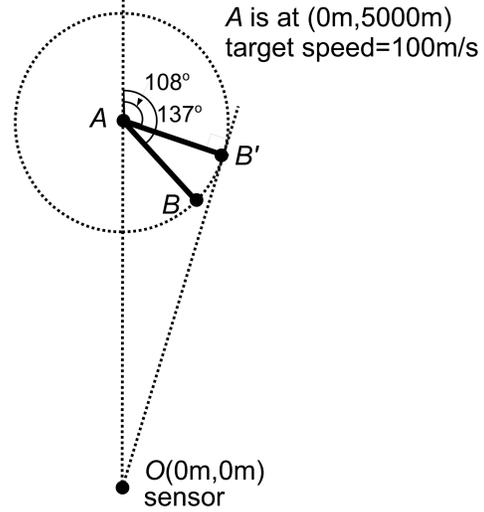


Fig. 7. The target trajectory \overline{AB} has the highest $\det[\mathcal{F}]$, and $\overline{AB'}$ has the largest bearing change.

The CRLB-based root mean square errors (RMSE) of the position and velocity are

$$\text{pos}_{\text{batch}}^{\text{CRLB}} = \sqrt{\mathbf{P}_{11}^{\text{CRLB}} + \mathbf{P}_{22}^{\text{CRLB}}} \quad (34)$$

$$\text{vel}_{\text{batch}}^{\text{CRLB}} = \sqrt{\mathbf{P}_{33}^{\text{CRLB}} + \mathbf{P}_{44}^{\text{CRLB}}} \quad (35)$$

The ILS RMSEs of the final position and velocity estimates are computed based on 200 Monte Carlo runs for each of the above batches. They are given by

$$\text{pos}_{\text{batch}}^{\text{RMSE}} = \sqrt{\frac{1}{N} \sum_{i=1}^N [\text{pos}_{\text{batch},i}^{\text{err}}]^2} \quad (36)$$

$$\text{vel}_{\text{batch}}^{\text{RMSE}} = \sqrt{\frac{1}{N} \sum_{i=1}^N [\text{vel}_{\text{batch},i}^{\text{err}}]^2} \quad (37)$$

where i is the run index, $N = 200$ is the number of runs, and

$$\text{pos}_{\text{batch},i}^{\text{err}} = \sqrt{[\hat{x} - x]^2 + [\hat{y} - y]^2} \quad (38)$$

$$\text{vel}_{\text{batch},i}^{\text{err}} = \sqrt{[\hat{\dot{x}} - \dot{x}]^2 + [\hat{\dot{y}} - \dot{y}]^2} \quad (39)$$

where \hat{x} , \hat{y} , $\hat{\dot{x}}$ and $\hat{\dot{y}}$ are the estimated target position and velocity in x and y coordinates, respectively, x , y , \dot{x} and \dot{y} are the true target positions and velocities, respectively.

To evaluate the consistency of the estimates obtained via the ILS with the CRLB (i.e. its statistical efficiency), the normalized estimation error squared (NEES) [2] is evaluated. The full state NEES for N Monte Carlo runs is

$$\bar{\epsilon} = \frac{1}{N} \sum_{i=1}^N (\hat{\mathbf{x}}_i - \mathbf{x})' \mathcal{F} (\hat{\mathbf{x}}_i - \mathbf{x}) \quad (40)$$

where i is the run index.

Table I presents the ILS RMSEs of the position and velocity estimates versus the CRLBs. Fig. 8 shows

TABLE I
Estimate RMSEs versus CRLB

Batch size	Target heading (°)	Position		Velocity		NEES
		CRLB (m)	ILS RMSE (m)	CRLB (m/s)	ILS RMSE (m/s)	
n = 10	45	1961.0	2893.9	139.7	187.3	5.04
	50	1735.2	1962.3	124.4	130.3	4.17
	55	1548.3	1848.4	111.8	123.9	4.13
	60	1390.9	1498.2	101.2	107.6	4.23
	80	949.1	1020.4	72.0	77.5	4.01
	100	682.6	722.52	54.9	45.2	4.09
	140	502.6	535.1	45.3	46.9	4.31
n = 20	140	451.1	478.2	43.3	44.5	3.70
	120	2516.9	2680.5	118.1	122.8	4.60
	100	1994.3	2001.3	93.8	90.0	4.19
	80	1642.5	1653.3	77.5	77.5	4.28
	60	1195.5	1223.1	56.9	56.9	4.29
	40	731.3	753.7	35.6	37.2	4.24
	20	486.0	487.7	24.6	24.7	3.96
n = 30	140	333.5	343.9	17.8	18.3	3.81
	120	232.9	234.5	13.4	13.9	4.02
	100	169.9	174.2	10.9	11.4	4.06
	80	2565.2	3011.2	87.0	95.0	4.79
	60	1908.5	1970.0	64.8	64.8	4.02
	40	1511.0	1553.3	51.4	52.2	4.26
	20	902.5	915.8	31.0	31.8	4.23
n = 60	140	547.3	565.8	19.2	20.1	3.79
	120	358.2	361.9	12.9	13.2	4.20
	100	238.9	248.9	9.0	9.2	4.39
	80	157.7	160.6	6.4	6.5	3.70
	60	101.0	103.3	4.6	4.7	4.20
	40	2893.1	3066.2	48.4	62.6	4.50
	20	1917.7	2102.4	32.1	33.5	4.24

the ILS RMSEs of position estimates versus CRLBs in graph form, where n in legends ILS- n and CRLB- n stands for the batch size. It can be seen that the estimates are very close to their CRLBs, except for the cases with marginal observability (e.g., $n = 10$ and heading $\theta = 45^\circ$). In these cases the errors are large, and the ILS does not yield a statistically efficient estimate. Upon examining the reason why the estimate in these cases had large errors compared to the CRLB, it was observed that the likelihood $\Lambda(\mathbf{x}; \mathbf{Z})$ is larger at the (bad) estimate than at the true value (i.e., better goodness of fit to the noisy data for the bad estimate). This is because of the noisiness of the likelihood function due to the combined effect of the small number of measurements and the marginal observability.

The NEES is also shown in Table I. The 95% of probability region upper limit for a 800 degrees of freedom ($N = 200$, the size of \mathbf{x} is 4) chi-square random

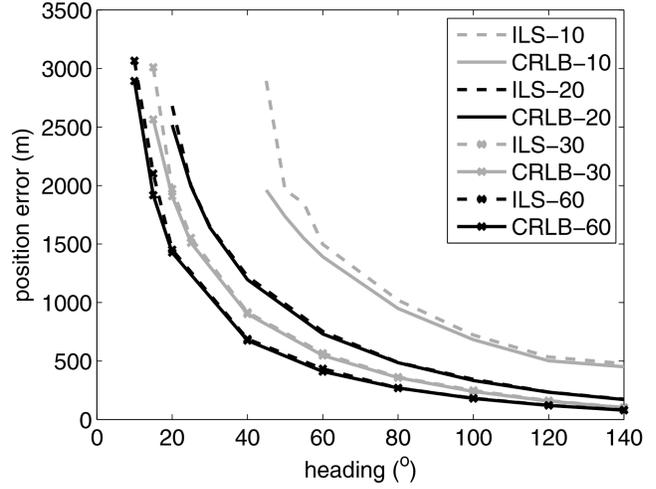


Fig. 8. Comparison of the ILS position estimation RMSE with the CRLB for $n = 10, 20, 30$ and 60 .

variable is 867. Dividing by $N = 200$, the NEES $\bar{\epsilon}$ should be less than 4.33. We carry out a test between the hypotheses

$$H_0: \mathcal{P} = \mathbf{P}^{\text{CRLB}} = \mathcal{F}^{-1} \quad (41)$$

$$H_1: \mathcal{P} > \mathbf{P}^{\text{CRLB}} = \mathcal{F}^{-1} \quad (42)$$

where \mathcal{P} is the actual covariance of the ML estimator. With the 95% probability region of $p(\bar{\epsilon} | H_0)$ the test accepts H_0 , i.e., it rejects H_1 at 5% significance level if

$$\bar{\epsilon} < 4.33 \quad (43)$$

In Table I, only 4 cases of the 35 cases considered do not satisfy (43). This shows that the ML estimate obtained via ILS yields results consistent with the CRLB in most of the cases. The four inconsistent cases (the first case in each batch category) are inserted on purpose, so that one can find the estimator's limitation through the NEES. We reduced the heading θ in each batch category until the NEES exceeds 4.33. It can be seen that the NEES exceeded 4.33 at $\theta = 45^\circ$ when $n = 10$, whereas this occurs at $\theta = 10^\circ$ when $n = 60$. Thus, the region of the ILS where the performance is consistent with the CRLB increases with the batch size.

Therefore the estimator's actual covariance is equal to the CRLB (with the exceptions noted above), i.e., the estimator presented is *statistically efficient*. This is in accordance to the well known property of the ML estimator that it is asymptotically efficient, i.e., for large n its covariance tends to the bound. In the present problem, this property holds for all but the first case (small number of measurements and marginal observability) from each group from Table I.

Fig. 9 shows the position error ellipses based on CRLB on the three target trajectories with heading 60° , 100° and 140° . The ellipses are drawn at time 10 s, 20 s, 30 s and 60 s, which correspond to batch sizes of $n = 10, 20, 30$ and 60 , respectively. From the orientation of the ellipses, we can observe that main position error is along

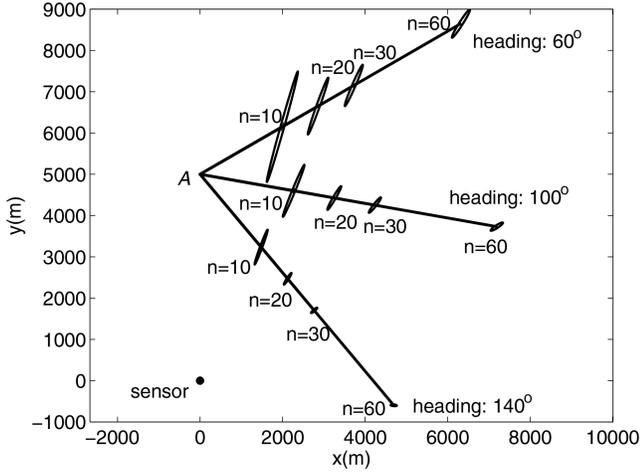


Fig. 9. Target position error ellipses based on CRLB. Three trajectories with heading 60° , 100° and 140° are shown. The position error ellipses are drawn for $n = 10, 20, 30$ and 60 .

the bearing line, and the cross-bearing error is relatively small. This is reasonable and commonly occurs in the BOT estimation problem.

V. RECURSIVE STATE ESTIMATION

A. The Fusion Architecture

The recursive state estimation updates the target state as measurements are received. It can be seen from Fig. 1 that out-of-sequence measurements (OOSM) occur due to the acoustic propagation delay.

The OOSM problem is also referred to as “negative-time measurement update” problem, namely, the state emission time, t_j^{e2} , corresponding to the latest measurement at t_k^{s2} is earlier than the latest state updating time, t_i^{e1} , namely $t_j^{e2} < t_i^{e1}$. The prediction step in the in-sequence estimation becomes a retrodiction for the OOSM. The OOSM problem has been extensively studied [5]. The simplest approach performs an approximate retrodiction by neglecting the process noise [5]. This approach is referred to as Algorithm C in [5]. Algorithms B1 and A1 were proposed to solve the one-step-lag OOSM by considering the process noise [9] [3], and they give an approximate and the exact solutions, respectively. They were further developed to the algorithms B/1 and A/1 for solving the l -step-lag OOSM ($l > 1$) in a single step [4].

The existing OOSM algorithms mentioned above assume that retrodiction time is known. However, the retrodiction time is the acoustic signal emission time in our problem. This is unknown to the observer and depends on the state of the target according to the following propagation delay constraint

$$t_j^{e2} = t_k^{s2} - \delta_{j,k} \quad (44)$$

where

$$\delta_{j,k} = \frac{r_{j,k}}{c^p} \quad (45)$$

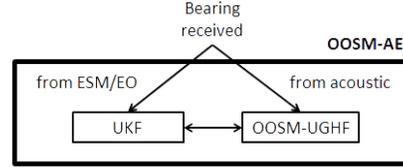


Fig. 10. OOSE-AE fusion architecture

is the propagation delay, c^p is the signal propagation speed in the medium, and $r_{j,k}$, which depends on the state (at the emission time), is the distance from the target at time t_j^{e2} to the sensor at time t_k^{s2} . This leads to an implicit constraint in the state transition model.

Recently, we have formulated an implicit-constraint dynamic estimation problem using a Gauss-Helmert model (GHM), and presented an unscented Gauss-Helmert filter (UGHF) [23] [24] to solve this problem. The UGHF works only with in-sequence measurements. The development for the OOSM-UGHF is one of the main contributions of this paper.

The recursive estimation problem in this paper is to estimate the target state with fusion of in-sequence bearings from the ESM/EO sensor (s_1) and out-of-sequence bearings from the acoustic sensor (s_2). The algorithm is called out-of-sequence measurement fusion for acoustic and ESM/EO sensors (OOSM-AE). Its architecture is shown in Fig. 10. State estimation for the bearings from s_1 will be performed by a unscented Kalman filter (UKF), which will be given next. For s_2 , a new OOSM-UGHF will be developed and described in Section VI.

B. The Model for the Recursive Estimation with Non-Delayed Bearings

The state estimation using the ESM/EO bearings is straightforward as the measurements arrive in-sequence and no propagation delay needs to be taken into consideration. The problem is formulated based on the nearly CV state model (or WNA—white noise acceleration). The target state, with dimension 4, is defined as

$$\mathbf{x}^4(t_k^{s1}) = [x(t_k^{s1}) \ y(t_k^{s1}) \ \dot{x}(t_k^{s1}) \ \dot{y}(t_k^{s1})]' \quad (46)$$

where t_k^{s1} is the signal reception (or sensor) time by the ESM/EO sensor s_1 at time cycle k . Since the propagation delay is negligible for s_1 , the target signal emission time t_i^{e1} is equal to t_k^{s1} . The state transition model is³

$$\mathbf{x}^4(t_k^{s1}) = \mathbf{F}(t_k^{s1}, t_{k-1}^{s1}) \mathbf{x}^4(t_{k-1}^{s1}) + \mathbf{v}^4(t_k^{s1}, t_{k-1}^{s1}) \quad (47)$$

where the transition matrix is

$$\mathbf{F}(t_k^{s1}, t_{k-1}^{s1}) = \begin{bmatrix} 1 & 0 & T_{k,k-1} & 0 \\ 0 & 1 & 0 & T_{k,k-1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (48)$$

³Here it is assumed for simplicity that the measurements arriving at t_{k-1} and t_k are both from sensor s_1 .

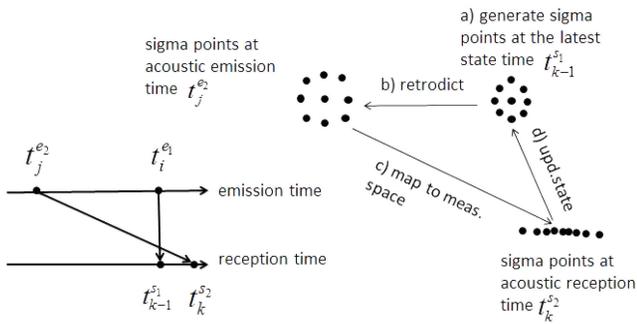


Fig. 11. OOSM-UGHF

and

$$T_{k,k-1} = t_k^{s1} - t_{k-1}^{s1} \quad (49)$$

with \mathbf{v}^4 the zero-mean process noise⁴ (WNA) for the interval $(t_{k-1}^{s1}, t_k^{s1}]$. The discretized white noise acceleration (DWNA) model [2] has covariance

$$E[\mathbf{v}^4(\cdot)\mathbf{v}^4(\cdot)'] = \mathbf{Q}^4(t_k^{s1}, t_{k-1}^{s1})$$

$$= \begin{bmatrix} \frac{T_{k,k-1}^3}{3} & 0 & \frac{T_{k,k-1}^2}{2} & 0 \\ 0 & \frac{T_{k,k-1}^3}{3} & 0 & \frac{T_{k,k-1}^2}{2} \\ \frac{T_{k,k-1}^2}{2} & 0 & T_{k,k-1} & 0 \\ 0 & \frac{T_{k,k-1}^2}{2} & 0 & T_{k,k-1} \end{bmatrix} q \quad (50)$$

where q is the power spectral density (PSD) of the (acceleration) process noise (same for x and y , and assumed independent between the coordinates). The measurement model is given by

$$z(t_k^{s1}) = h^4[\mathbf{x}^4(t_k^{s1})] = \tan^{-1} \left[\frac{x(t_k^{s1}) - x^s(t_k^{s1})}{y(t_k^{s1}) - y^s(t_k^{s1})} \right] + w(t_k^{s1}) \quad (51)$$

where $x^s(t_k^{s1})$ and $y^s(t_k^{s1})$ are the sensor positions at time t_k^{s1} in the x and y coordinates respectively, $w(t_k^{s1})$ is zero-mean white Gaussian measurement noise with variance $R(t_k^{s1})$, assumed independent of the process noise.

The unscented Kalman filter (UKF) is used to estimate the state as in [15].

VI. RECURSIVE STATE ESTIMATION WITH DELAYED BEARINGS

An out-of-sequence-measurement filter is required for the bearings from the acoustic sensor s_2 . It can be seen in Fig. 11 that an acoustic measurement received at time t_k^{s2} corresponds to the target state at emission time t_j^{e2} , which is earlier than the latest state assumed to have been updated by the ESM/EO sensor at time $t_{k-1}^{s1} = t_i^{e1}$. The problem is then to update the state estimate $\hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_{k-1}^{s1})$ with the acoustic measurement $z(t_k^{s2})$. The main challenge of this problem compared to the existing

OOSM approaches is that the time t_j^{e2} is unknown, and it needs to be estimated together with the kinematic state.

Instead of the first-order Taylor linearization used in the existing OOSM algorithms [4] [5], the unscented transform is used in the above mentioned problem. This consists of the following steps:

- Retrodict the state from time $t_i^{e1} = t_{k-1}^{s1}$ to the (unknown) emission time t_j^{e2} (to be estimated) corresponding to the sensor time t_k^{s2} . The state estimate before retrodiction is $\hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_{k-1}^{s1})$, and the retrodicted state is $\hat{\mathbf{x}}^5(t_j^{e2} | t_{k-1}^{s1})$. The latter, defined below in (52), includes the acoustic emission time. This step is illustrated through a) and b) in Fig. 11.
- Update the state estimate $\hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_{k-1}^{s1})$ to $\hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_k^{s2})$ with the acoustic OOSM $z(t_k^{s2})$. This step is illustrated through c) and d) in Fig. 11.

The algorithm details are presented next.

A. State Retrodiction

The retrodiction that has to be done to the emission time t_j^{e2} (unknown to the observer) is subject to the propagation delay constraint described in (44). To estimate the retrodicted target kinematic information and the emission time t_j^{e2} simultaneously, the following augmented state is defined:

$$\mathbf{x}^5(t_j^{e2}) = [x(t_j^{e2}) \ y(t_j^{e2}) \ \dot{x}(t_j^{e2}) \ \dot{y}(t_j^{e2}) \ t_j^{e2}]' \quad (52)$$

Obviously, the positions $x(t_j^{e2})$, $y(t_j^{e2})$ and the time t_j^{e2} depend on each other, and this leads to the retrodicted state $\hat{\mathbf{x}}^5(t_j^{e2} | t_{k-1}^{s1})$ and the latest state estimate $\hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_{k-1}^{s1})$ to have an implicit relationship. The Gauss-Helmert transition model [23] [24], which handles such implicit relationships, is then used for retrodiction. This is described by

$$\mathbf{g}[\mathbf{x}^5(t_j^{e2}), \mathbf{x}^4(t_{k-1}^{s1})] + \mathbf{v}^5(t_{k-1}^{s1}, t_j^{e2}) = \mathbf{0}_5 \quad (53)$$

where $\mathbf{g}[\cdot]$ is the Gauss-Helmert implicit state transition function, which combines the target motion constraints and the delay constraint between $\mathbf{x}^5(t_j^{e2})$ of dimension 5 and $\mathbf{x}^4(t_{k-1}^{s1})$ of dimension 4, and $\mathbf{0}_5$ is the zero vector of dimension 5. Assuming the target motion follows a WNA motion, $\mathbf{g}[\cdot]$ is given by

$$\mathbf{g}[\cdot] = [g_1(\cdot) \ g_2(\cdot) \ g_3(\cdot) \ g_4(\cdot) \ g_5(\cdot)]' \quad (54)$$

where

$$g_1 = x(t_j^{e2}) - x(t_{k-1}^{s1}) - \dot{x}(t_{k-1}^{s1})T_{j,k-1} \quad (55)$$

$$g_2 = y(t_j^{e2}) - y(t_{k-1}^{s1}) - \dot{y}(t_{k-1}^{s1})T_{j,k-1} \quad (56)$$

$$g_3 = \dot{x}(t_j^{e2}) - \dot{x}(t_{k-1}^{s1}) \quad (57)$$

$$g_4 = \dot{y}(t_j^{e2}) - \dot{y}(t_{k-1}^{s1}) \quad (58)$$

$$g_5 = t_j^{e2} + \frac{r_{j,k}}{c_p} - t_k^{s2} \quad (59)$$

and

$$T_{j,k-1} = t_j^{e2} - t_{k-1}^{s1} < 0 \quad (60)$$

⁴The process noise arguments are shown in the same manner as for the state transition matrix.

$$r_{j,k} = \sqrt{[x(t_j^{e2}) - x^s(t_k^{s2})]^2 + [y(t_j^{e2}) - y^s(t_k^{s2})]^2} \quad (61)$$

Note that (59) is the equation that connects the emission time and target location to the corresponding sensor reception time.

The process noise \mathbf{v}^5 in (53) is modeled as zero-mean Gaussian. Based on the DWNA model [5], its covariance is

$$\mathbf{Q}^5(t_j^{e2}, t_{k-1}^{s1}) = \begin{bmatrix} \frac{|T_{j,k-1}|^3}{3}q & 0 & \frac{T_{j,k-1}^2}{2}q & 0 & 0 \\ 0 & \frac{|T_{j,k-1}|^3}{3}q & 0 & \frac{T_{j,k-1}^2}{2}q & 0 \\ \frac{T_{j,k-1}^2}{2}q & 0 & |T_{j,k-1}|q & 0 & 0 \\ 0 & \frac{T_{j,k-1}^2}{2}q & 0 & |T_{j,k-1}|q & 0 \\ 0 & 0 & 0 & 0 & q_\delta \end{bmatrix} \quad (62)$$

where q is as in (50), and q_δ is the variance of the process noise in the delay.

The algorithm used for retrodiction is the UGHF [24] [23], which obtains the retrodicted state iteratively through a Gauss-Newton algorithm. Given $\hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_{k-1}^{s1})$ and its error covariance $\mathbf{P}^4(t_{k-1}^{s1} | t_{k-1}^{s1})$, the sigma points and their corresponding weights are

$$\{\{\hat{\mathbf{x}}^{4,m}(t_{k-1}^{s1} | t_{k-1}^{s1})\}, \{w^m\}\} = \text{SigmaPts}[\hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_{k-1}^{s1}), \mathbf{P}^4(t_{k-1}^{s1} | t_{k-1}^{s1}), \kappa] \quad (63)$$

with⁵

$$\hat{\mathbf{x}}^{4,0}(t_{k-1}^{s1} | t_{k-1}^{s1}) = \hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_{k-1}^{s1}) \quad (64)$$

$$\hat{\mathbf{x}}^{4,m}(t_{k-1}^{s1} | t_{k-1}^{s1}) = \hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_{k-1}^{s1}) \quad (65)$$

$$+ \left[\sqrt{(4 + \kappa)\mathbf{P}^4(t_{k-1}^{s1} | t_{k-1}^{s1})} \right]_{|m|} \quad m = 1, \dots, 4 \quad (66)$$

$$\hat{\mathbf{x}}^{4,m}(t_{k-1}^{s1} | t_{k-1}^{s1}) = \hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_{k-1}^{s1}) - \left[\sqrt{(4 + \kappa)\mathbf{P}^4(t_{k-1}^{s1} | t_{k-1}^{s1})} \right]_{|m|} \quad m = -4, \dots, -1$$

$$w^0 = \frac{\kappa}{4 + \kappa} \quad (67)$$

$$w^m = \frac{1}{2(4 + \kappa)} \quad |m| = 1, \dots, 4 \quad (68)$$

where $m = -4, \dots, 4$, is the sigma point index, $\left[\sqrt{(4 + \kappa)\mathbf{P}^4(t_{k-1}^{s1} | t_{k-1}^{s1})} \right]_{|m|}$ indicates the $|m|$ th column

⁵Since $\hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_{k-1}^{s1})$ has dimension 4, there are 9 sigma points [15].

of the matrix $\left[\sqrt{(4 + \kappa)\mathbf{P}^4(t_{k-1}^{s1} | t_{k-1}^{s1})} \right]$, and κ is a scalar that determines the spread of sigma points. Each sigma point is retrodicted from the previous target time t_{k-1}^{s1} to an unknown time $(t_j^{e2})^m$. The problem is then to solve

$$\mathbf{g}[\hat{\mathbf{x}}^{5,m}(t_j^{e2} | t_{k-1}^{s1}), \hat{\mathbf{x}}^{4,m}(t_{k-1}^{s1} | t_{k-1}^{s1})] = \mathbf{0}_5 \quad m = -4, \dots, 4 \quad (69)$$

Note that the process noise is not taken into consideration in the OOSM algorithm C.

The Gauss-Newton algorithm is applied to obtain the points $\hat{\mathbf{x}}^{5,m}(t_j^{e2} | t_{k-1}^{s1})$ iteratively. The iteration procedure (with index n) for the m th sigma point is

$$\begin{aligned} [\hat{\mathbf{x}}^{5,m}(t_j^{e2} | t_{k-1}^{s1})]^n &= [\hat{\mathbf{x}}^{5,m}(t_j^{e2} | t_{k-1}^{s1})]^{n-1} \\ &+ \mathbf{A}^{-1} \mathbf{g}[[\hat{\mathbf{x}}^{5,m}(t_j^{e2} | t_{k-1}^{s1})]^{n-1}, \hat{\mathbf{x}}^{4,m}(t_{k-1}^{s1} | t_{k-1}^{s1})] \end{aligned} \quad (70)$$

where \mathbf{A} (without arguments, for conciseness) is the Jacobian matrix given by

$$\begin{aligned} \mathbf{A} &= \frac{\partial \mathbf{g}[[\hat{\mathbf{x}}^{5,m}(t_j^{e2} | t_{k-1}^{s1})]^n, \hat{\mathbf{x}}^{4,m}(t_{k-1}^{s1} | t_{k-1}^{s1})]}{\partial [\hat{\mathbf{x}}^{5,m}(t_j^{e2} | t_{k-1}^{s1})]^n} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & -\dot{x}^m(t_{k-1}^{s1} | t_{k-1}^{s1}) \\ 0 & 1 & 0 & 0 & -\dot{y}^m(t_{k-1}^{s1} | t_{k-1}^{s1}) \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \frac{x_{j,k}^r}{r_{j,k}^{cP}} & \frac{y_{j,k}^r}{r_{j,k}^{cP}} & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (71)$$

and

$$x_{j,k}^r \triangleq [x^m(t_j^{e2} | t_{k-1}^{s1})]^n - x^s(t_k^{s2}) \quad (72)$$

$$y_{j,k}^r \triangleq [y^m(t_j^{e2} | t_{k-1}^{s1})]^n - y^s(t_k^{s2}) \quad (73)$$

$$r_{j,k} \triangleq \sqrt{(x_{j,k}^r)^2 + (y_{j,k}^r)^2} \quad (74)$$

The Gauss-Newton algorithm described in (70) is quadratically convergent to the unique solution when a target is not approaching the sensor with radial speed c^P , if we assume the initial point is reasonably close to the solution [24].

The initial value of the m th sigma point $[\hat{\mathbf{x}}^{5,m}(t_j^{e2} | t_{k-1}^{s1})]^0$ for the iteration (70) is computed as

$$[x^m(t_j^{e2} | t_{k-1}^{s1})]^0 = x^m(t_{k-1}^{s1} | t_{k-1}^{s1}) + \dot{x}^m(t_{k-1}^{s1} | t_{k-1}^{s1})[\Delta(t_j^{e2})]^0 \quad (75)$$

$$[y^m(t_j^{e2} | t_{k-1}^{s1})]^0 = y^m(t_{k-1}^{s1} | t_{k-1}^{s1}) + \dot{y}^m(t_{k-1}^{s1} | t_{k-1}^{s1})[\Delta(t_j^{e2})]^0 \quad (76)$$

$$[\dot{x}^m(t_j^{e2} | t_{k-1}^{s1})]^0 = \dot{x}^m(t_{k-1}^{s1} | t_{k-1}^{s1}) \quad (77)$$

$$[\dot{y}^m(t_j^{e2} | t_{k-1}^{s1})]^0 = \dot{y}^m(t_{k-1}^{s1} | t_{k-1}^{s1}) \quad (78)$$

$$[(t_j^{e2})^m]^0 = t_k^{s2} - \delta_{j,k} \quad (79)$$

where

$$[\Delta(t_j^{e2})]^0 = [(t_j^{e2})^m]^0 - t_{k-1}^{s1} \quad (80)$$

$$\delta_{j,k} \approx r_{k-1}^m / c^p \quad (81)$$

and r_{k-1}^m is the distance between the target position estimate and the sensor at time t_{k-1}^{s1} .

The retrodicted state $\hat{\mathbf{x}}^5(t_j^{e2} | t_{k-1}^{s1})$ and its error covariance $\mathbf{P}^5(t_j^{e2} | t_{k-1}^{s1})$ are then computed from the following weighted sums of the retrodicted sigma points

$$\hat{\mathbf{x}}^5(t_j^{e2} | t_{k-1}^{s1}) = \sum_{m=-4}^4 w^m \hat{\mathbf{x}}^{5,m}(t_j^{e2} | t_{k-1}^{s1}) \quad (82)$$

$$\mathbf{P}^5(t_j^{e2} | t_{k-1}^{s1}) \approx \sum_{m=-4}^4 w^m \tilde{\mathbf{x}}^{5,m}(t_j^{e2} | t_{k-1}^{s1}) (\tilde{\mathbf{x}}^{5,m}(t_j^{e2} | t_{k-1}^{s1}))' \quad (83)$$

where

$$\tilde{\mathbf{x}}^{5,m}(t_j^{e2} | t_{k-1}^{s1}) = \hat{\mathbf{x}}^{5,m}(t_j^{e2} | t_{k-1}^{s1}) - \hat{\mathbf{x}}^5(t_j^{e2} | t_{k-1}^{s1}) \quad (84)$$

with $m = -4, \dots, 4$.

B. State Update

This step updates $\hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_{k-1}^{s1})$ to $\hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_k^{s2})$ by the OOSM $z(t_k^{s2})$ —it fuses the latter into the former. Note that the sigma points of $\hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_{k-1}^{s1})$ have been generated in (63).

According to the linear minimum mean square error (LMMSE) estimator [2], the estimate $\hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_k^{s2})$ and its error covariance $\mathbf{P}^4(t_{k-1}^{s1} | t_k^{s2})$ are given by

$$\hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_k^{s2}) = \hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_{k-1}^{s1}) + \mathbf{P}_{xz} P_{zz}^{-1} [z(t_k^{s2}) - \hat{z}(t_k^{s2})] \quad (85)$$

$$\mathbf{P}^4(t_{k-1}^{s1} | t_k^{s2}) = \mathbf{P}^4(t_{k-1}^{s1} | t_{k-1}^{s1}) - \mathbf{P}_{xz} P_{zz}^{-1} \mathbf{P}'_{xz} \quad (86)$$

The expected measurement $\hat{z}(t_k^{s2})$, based on the retrodicted state $\hat{\mathbf{x}}^{5,m}(t_j^{e2} | t_{k-1}^{s1})$, is

$$\hat{z}(t_k^{s2}) = \sum_{m=-4}^4 w^m \hat{z}^m(t_k^{s2}) \quad (87)$$

where

$$\begin{aligned} \hat{z}^m(t_k^{s2}) &= h^5[\hat{\mathbf{x}}^{5,m}(t_j^{e2} | t_{k-1}^{s1})] \\ &= \tan^{-1} \left[\frac{x^m(t_j^{e2} | t_{k-1}^{s1}) - x^s(t_k^{s2})}{y^m(t_j^{e2} | t_{k-1}^{s1}) - y^s(t_k^{s2})} \right] \end{aligned} \quad (88)$$

The variance P_{zz} of the innovation and the covariance \mathbf{P}_{xz} between the state to be estimated and the measurement are computed as

$$P_{zz} = \sum_{m=-4}^4 w^m [\hat{z}^m(t_k^{s2})]^2 + R(t_k^{s2}) \quad (89)$$

$$\mathbf{P}_{xz} = \sum_{m=-4}^4 w^m \tilde{\mathbf{x}}^{4,m}(t_{k-1}^{s1} | t_{k-1}^{s1}) \hat{z}^m(t_k^{s2}) \quad (90)$$

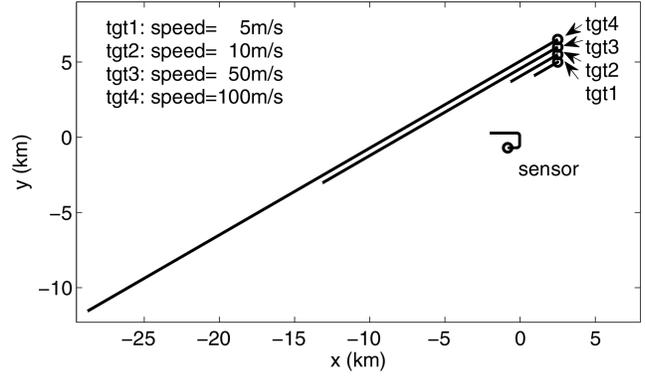


Fig. 12. Test scenarios. Initial locations of the targets and the maneuvering sensor platform are shown as “o.” The stationary platform is located at (0,0)

where

$$\tilde{\mathbf{x}}^{4,m}(t_{k-1}^{s1} | t_{k-1}^{s1}) = \hat{\mathbf{x}}^{4,m}(t_{k-1}^{s1} | t_{k-1}^{s1}) - \hat{\mathbf{x}}^4(t_{k-1}^{s1} | t_{k-1}^{s1}) \quad (91)$$

$$\tilde{z}^m(t_k^{s2}) = \hat{z}^m(t_k^{s2}) - \hat{z}(t_k^{s2}) \quad (92)$$

The OOSM-UGHF does not create new states, it only updates the state generated by the bearing from s_1 before.

VII. SIMULATION RESULTS FOR RECURSIVE ESTIMATION

Simulation results are given to demonstrate the new algorithm's performance. The conventional BOT approach is also evaluated using the same simulation data. Two sensor platform scenarios are used in the simulation tests:

- **Maneuvering (M):** It has three legs linked by two 90° turns with turn rate $3^\circ/\text{s}$ shown in Fig. 12. The platform speed is 10 m/s throughout the whole path. It moves to the east for 60 s, spends 30 s to make a 90° left turn, and moves to the north for 60 s. It then makes a 90° left turn, and moves towards the west for 180 s. The total duration is 360 s.
- **Stationary (S):** The platform stays at position (0 m, 0 m) for 360 s.

An ESM sensor and an acoustic sensor are deployed on the platform to detect target bearings. The two sensors are not synchronized. Their sampling intervals and initial detection times are different. The ESM sensor is initiated at time 0 s with sampling interval 1s, whereas, the acoustic sensor is initiated at time 0.2 s with sampling interval 2 s. The 2:1 ratio of the sampling intervals is determined by the assumed reasonable sampling times of ESM and acoustic sensors. It can be set to other values based on real applications. The measured bearing errors of the ESM and the acoustic sensors are zero-mean white Gaussian with standard deviations $\sigma_b = 1^\circ$. We assume that both sensors have no bearing detection during platform turns (total missed detection duration is

60 s for the maneuvering platform scenarios). This assumption is valid for most real applications. Typically, during a turn the pointing of the sensors is not known accurately.

Four targets moving at constant speeds of 5 m/s, 10 m/s, 50 m/s and 100 m/s, respectively, are shown in Fig. 12 (the actual trajectories have process noise, as discussed in the sequel). The state estimation starts 50 s after the targets move from their initial positions and the first bearing is from the ESM, so the acoustic signal can be guaranteed to reach the sensor platform when the estimation starts. This means that the targets are at their initial points at time -50 s, and the sensor platform is at its initial point at time 0 s. The estimation starts at time 0 s.

The algorithms used in the simulation are:

- **OOSM-AE**: The acoustic-ESM fusion algorithm proposed in this paper. The OOSM-UGHF is used for the bearings from the acoustic sensor, and the UKF is used for the bearings from the ESM. It works for both stationary and moving (maneuvering or nonmaneuvering) platform.
- **UKF-E**: A UKF to estimate state based on the ESM bearings only. The acoustic bearings are regarded as “expired” information and discarded. This algorithm is the conventional BOT approach, which works for maneuvering platform only.

The initial state estimate is

$$\hat{\mathbf{x}}^4(t_0^{s_1}) = [r_0 \sin b_0 \quad r_0 \cos b_0 \quad \dot{x}_0 \quad \dot{y}_0]' \quad (93)$$

where $b_0 = b(t_0^{s_1})$ is the ESM measured bearing at time $t_0^{s_1} = 0$ s,

$$r_0 \sim \mathcal{N}(\hat{r}_0, \sigma_{r_0}^2) \quad (94)$$

$$\dot{x}_0 \sim \mathcal{N}(0, \sigma_{\dot{x}_0}^2) \quad (95)$$

$$\dot{y}_0 \sim \mathcal{N}(0, \sigma_{\dot{y}_0}^2) \quad (96)$$

with $\hat{r}_0 = 7500$ m is half of the detection range (assuming 15000 m), $\sigma_{r_0} = 2500$ m, and $\sigma_{\dot{x}_0} = \sigma_{\dot{y}_0} = 30$ m/s. The initial state error covariance is computed by [20]

$$\mathbf{P}^4(t_0^{s_1}) = \begin{bmatrix} P_{xx} & P_{xy} & 0 & 0 \\ P_{yx} & P_{yy} & 0 & 0 \\ 0 & 0 & \sigma_{\dot{x}_0}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{y}_0}^2 \end{bmatrix} \quad (97)$$

where

$$P_{xx} = (\hat{r}_0 \sigma_b \cos b_0)^2 + (\sigma_r \sin b_0)^2 \quad (98)$$

$$P_{yy} = (\hat{r}_0 \sigma_b \sin b_0)^2 + (\sigma_r \cos b_0)^2 \quad (99)$$

$$P_{xy} = P_{yx} = (\sigma_r^2 - \hat{r}_0^2 \sigma_b^2) \sin b_0 \cos b_0 \quad (100)$$

In other words, in each run we have a random initial state, which is in accordance with the Bayesian model (see, e.g. [2] Sec. 5.5).

The process noise PSD q in (50) is set to $0.01\text{m}^2/\text{s}^3$. Note that due to the presence of process noise, batch estimation is not applicable. The acoustic signal propagation speed c^p in the air is 344 m/s. The scalar κ in (63)–(68) is set to 1.

The UGHF-E performance is investigated below in several aspects.

A. Root Mean Square Errors

The estimated position root mean square errors (RMSE) obtained from 100 Monte Carlo runs versus time are displayed in Figs. 13–16 for the maneuvering platform, and Figs. 17–18 for the stationary platform. The overall and the last point position RMSEs for all the scenarios are given in Table II. The RMSEs of the UKF-E are not shown in this table for the stationary platform, because the targets are unobservable in this case. The overall position RMSE for a particular scenario is computed by

$$\text{pos}^{\text{RMSE}} = \sqrt{\frac{1}{NK} \sum_{i=1}^N \sum_{k=1}^K [\text{pos}_i^{\text{err}}(t_k^{s_1})]^2} \quad (101)$$

where i is the run index, $N = 100$ is the number of runs, $K = 360$ is the number of time cycles in the scenario, and

$$\text{pos}^{\text{err}}(t_k^{s_1}) = \sqrt{[\hat{x}(t_k^{s_1}) - x(t_k^{s_1})]^2 + [\hat{y}(t_k^{s_1}) - y(t_k^{s_1})]^2} \quad (102)$$

where $\hat{x}(t_k^{s_1})$ and $\hat{y}(t_k^{s_1})$ are the estimated target positions, and $x(t_k^{s_1})$ and $y(t_k^{s_1})$ are the true target positions.

It can be seen that the OOSM-AE clearly outperforms the UKF-E for the maneuvering platform scenarios. The overall accuracy improvement in terms of position RMSE reduction is from 69% to 77%, a significant improvement. For the slow moving targets (shown in Figs. 13–14), the UKF-E takes a longer time to converge. The UKF-E position RMSEs start to decrease at time 180 s (after the second turn), whereas the RMSE reduction of the OOSM-AE occurs around time 50 s, which is much earlier than for the UKF-E. For the fast moving targets (shown in Figs. 15–16), both algorithms converge fast at the beginning, but the UKF-E has larger errors after a while.

For the stationary platform (Figs. 17–18), the OOSM-AE provides reliable estimation, whereas the UKF-E diverges since BOT from a single fixed passive sensor is not observable.

We also observe that the OOSM-AE has better performance for the fast moving targets than the slow moving targets in both maneuvering and stationary platform scenarios. The reason for this is that the slow moving targets have lower bearing change rate. The information provided by these slowly changing bearings is limited when they are “buried in the noise,” and this results in marginal observability and slow convergence in the beginning. This effect is more serious for the stationary platform as its bearing change rate is even smaller than for the maneuvering platform.

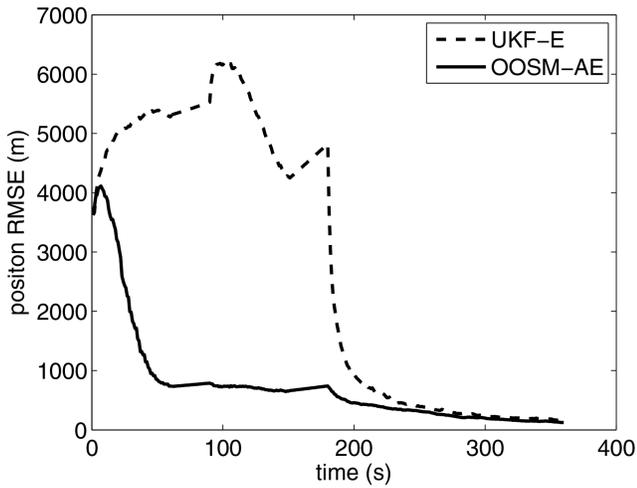


Fig. 13. Maneuvering platform: The estimated position RMSE versus time for the target with speed 5 m/s.

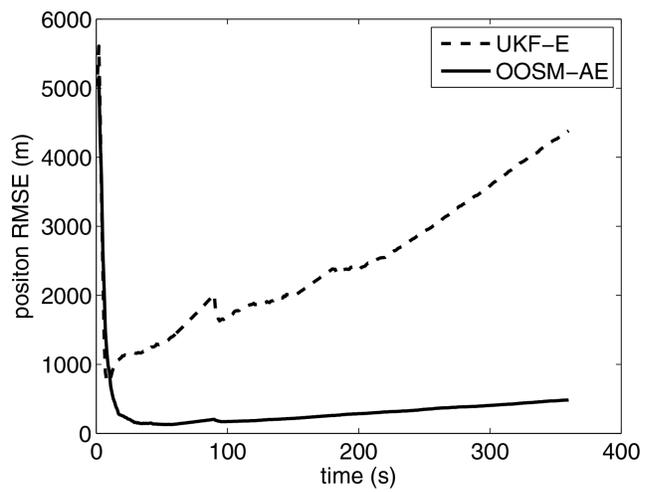


Fig. 16. Maneuvering platform: The estimated position RMSE versus time for the target with speed 100 m/s.

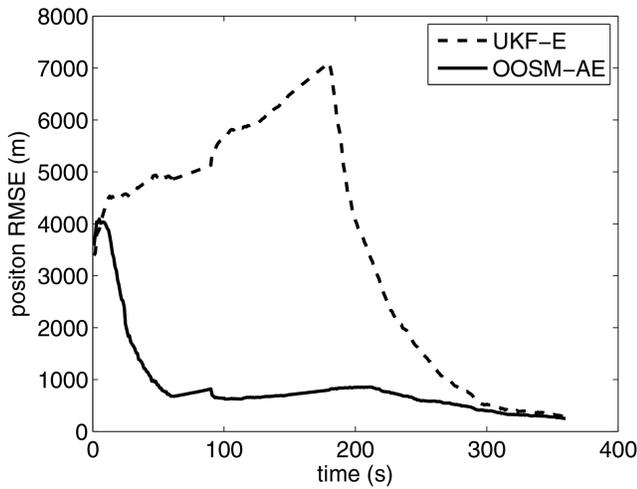


Fig. 14. Maneuvering platform: The estimated position RMSE versus time for the target with speed 10 m/s.

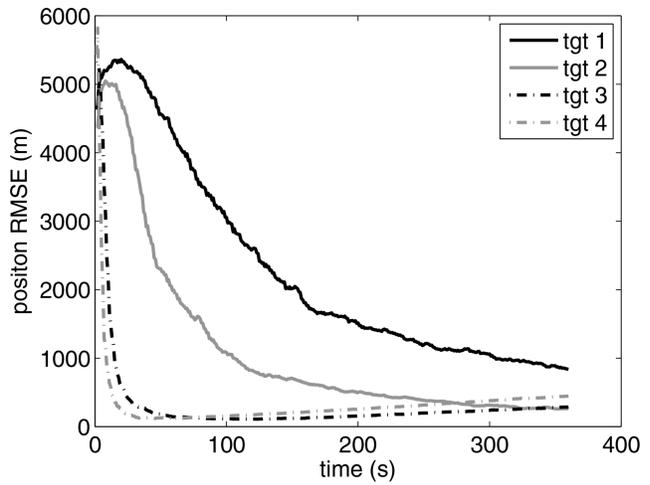


Fig. 17. Stationary platform: The OOSM-AE estimated position RMSE versus time for the targets with speeds of 5 m/s (tgt 1), 10 m/s (tgt 2), 50 m/s (tgt 3) and 100 m/s (tgt 4).

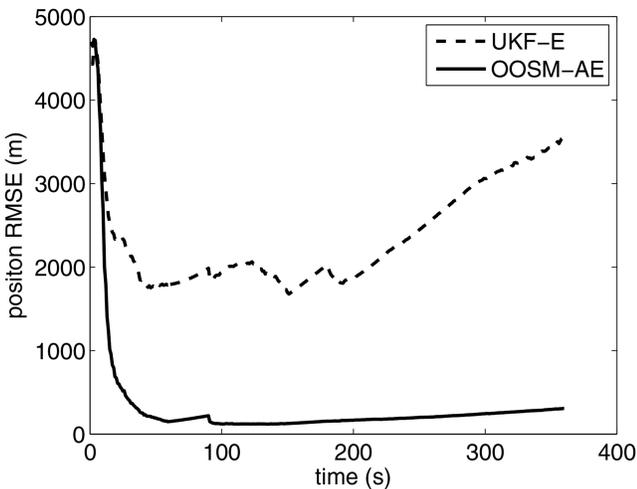


Fig. 15. Maneuvering platform: The estimated position RMSE versus time for the target with speed 50 m/s.

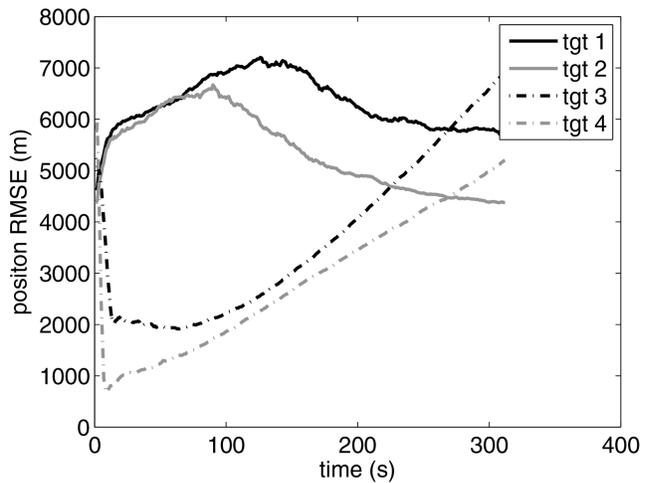


Fig. 18. Stationary platform: The UKF-E estimated position RMSE versus time for the four targets with speeds of 5 m/s (tgt 1), 10 m/s (tgt 2), 50 m/s (tgt 3) and 100 m/s (tgt 4).

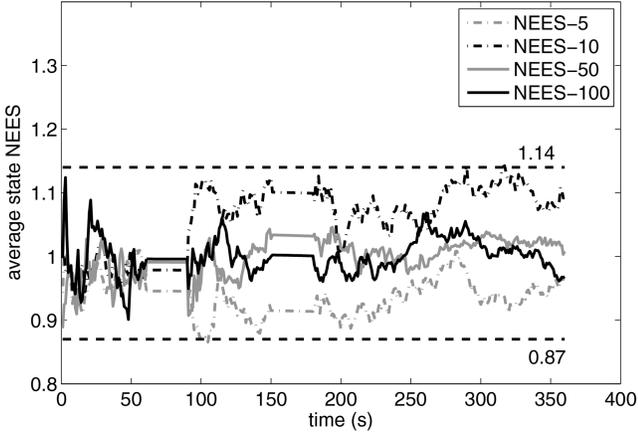


Fig. 19. The average state NEES (from 100 runs) for maneuvering platform scenarios with target speeds 5 m/s (NEES-5), 10 m/s (NEES-10), 50 m/s (NEES-50) and 100 m/s (NEES-100).

TABLE II
Position RMSEs for all the scenarios

Platform	Target speed (m/s)	Overall RMSE			Last point RMSE	
		OOSM-AE (m)	UKF-E (m)	Improvement (%)	OOSM-AE (m)	UKF-E (m)
M	5	1134.4	3694.5	69.3	124.2	156.3
	10	1152.4	4261.3	73.0	252.1	289.1
	50	752.5	2461.4	69.4	307.0	3540.7
	100	609.3	2673.5	77.2	483.9	4381.5
S	5	2724.2	—	—	836.2	—
	10	1708.2	—	—	261.3	—
	50	755.7	—	—	286.7	—
	100	624.2	—	—	445.0	—

B. Statistical Analysis: Consistency and Efficiency

To evaluate the consistency of OOSM-AE, the average normalized estimation error squared (NEES) is evaluated. The average state NEES at time t_k^s for N Monte Carlo runs is [2]

$$\bar{\epsilon}(t_k^s) = \frac{1}{n_{x^4} N} \sum_{i=1}^N \tilde{\mathbf{x}}_i^4(t_k^s)' \mathbf{P}^{-1}(t_k^s) \tilde{\mathbf{x}}_i^4(t_k^s) \quad (103)$$

where $\mathbf{P}(t_k^s)$ is the state error covariance computed by the OOSM-AE estimator, $n_{x^4} = 4$ is the state dimension, i is the run index, and

$$\tilde{\mathbf{x}}_i^4(t_k^s) = \mathbf{x}^4(t_k^s) - \hat{\mathbf{x}}_i^4(t_k^s) \quad (104)$$

The two-sided 95% probability region for a 400 degrees of freedom ($N = 100$, $n_{x^4} = 4$) chi-square random variable is [346.5, 457.3]. Dividing by 400, the average NEES should be in the interval [0.87, 1.14]. Figs. 19 and 20 show the average NEES versus time in the OOSM-AE for the maneuvering and stationary platform, respectively, where n in NEES-n stands for the target speed. It can be seen that the NEES for all test

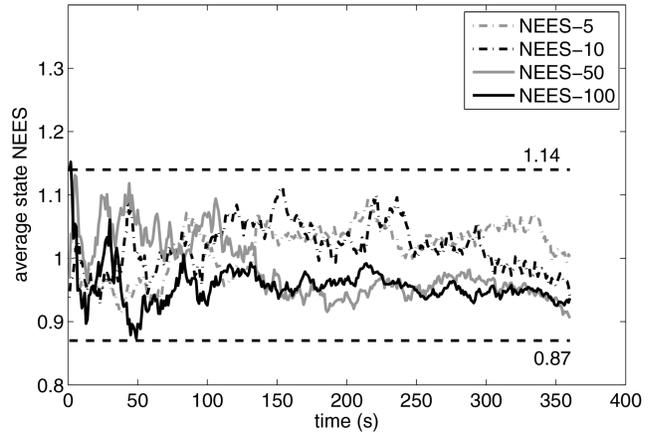


Fig. 20. The average state NEES (from 100 runs) for stationary platform scenarios with target speeds 5 m/s (NEES-5), 10 m/s (NEES-10), 50 m/s (NEES-50) and 100 m/s (NEES-100).

cases are within the range. This shows that OOSM-AE yields consistent estimation results.

Please note that the initial state for each run is randomly generated based on (93) with

$$r_0 \sim \mathcal{N}(r_0^{\text{true}}, \sigma_{r_0}^2) \quad (105)$$

$$\dot{x}_0 \sim \mathcal{N}(\dot{x}_0^{\text{true}}, \sigma_{\dot{x}_0}^2) \quad (106)$$

$$\dot{y}_0 \sim \mathcal{N}(\dot{y}_0^{\text{true}}, \sigma_{\dot{y}_0}^2) \quad (107)$$

where r_0^{true} , \dot{x}_0^{true} and \dot{y}_0^{true} are true values at time 0, and the error standard deviations are

$$\sigma_{r_0} = 0.2r_0^{\text{true}} \quad (108)$$

$$\sigma_{\dot{x}_0} = 0.2\dot{x}_0^{\text{true}} \quad (109)$$

$$\sigma_{\dot{y}_0} = 0.2\dot{y}_0^{\text{true}} \quad (110)$$

This setting prevents the initial state (randomly generated) from bias and apart from the ground truth significantly (which may cause divergency).

The efficiency of the OOSM-AE can be studied through the posterior CRLB (PCRLB) which is also called Bayesian CRLB (BCRLB). It is the inverse of the Bayesian information matrix (BIM), \mathbf{J} [21]. An estimator with state error $\tilde{\mathbf{x}}_i^4(t_k^s)$ is statistically efficient iff

$$\frac{1}{n_{x^4}} E[\tilde{\mathbf{x}}_i^4(t_k^s)' \mathbf{J}(t_k^s) \tilde{\mathbf{x}}_i^4(t_k^s)] = 1 \quad (111)$$

where $\mathbf{J}(t_k^s)$ is the BIM at time t_k^s . Since $\mathbf{P}^{-1}(t_k^s)$ is a good (run-specific) approximation of $\mathbf{J}(t_k^s)$ ⁶ (the former is conditioned on the measurements, while the latter is the average over all the measurements and states), the NEES (from 100 runs) can than be used to evaluate the estimator efficiency. It can be seen the NEES in Figs. 19 and 20 are within the 95% probability region.

⁶Based on our best knowledge, the BIM for the UGHF has not studied in literature. $\mathbf{P}^{-1}(t_k^s)$ is therefore used as the approximation of BIM.

VIII. CONCLUSIONS

This paper presented a new passive BOT approach through fusion of an ESM/EO and an acoustic sensor deployed on the same sensor platform. The OOSM-AE algorithm has been developed to estimate the target trajectory by utilizing the acoustic propagation delay which contains target range information. This approach avoids the requirement for platform maneuvers of the conventional BOT. The observability study conducted for this problem showed that the target state is completely observable when its bearing from the sensor platform is not a constant over time. Two algorithms, the ML estimator computed via ILS and OOSM-AE, were developed for batch and recursive estimations, respectively. Simulation results showed that the OOSM-AE can estimate the target trajectory effectively even from a stationary platform, and provides significant accuracy improvement (69%–77%) over the conventional BOT for the maneuvering platform cases considered. Statistical studies on consistency and efficiency were also conducted. The ML estimates obtained via ILS for a constant velocity target are statistically efficient, except for the case with too few measurements and marginal observability. The OOSM-AE yields consistent estimation results, and its average NEES is close 1. Thus, the new approach has the potential to enhance passive BOT capability significantly.

REFERENCES

- [1] Aidala, V. J.
Kalman filter behavior in bearings-only tracking applications,
IEEE Transactions on Aerospace and Electronic Systems, 15, 1 (Jan. 1979), 29–39.
- [2] Bar-Shalom, Y., Li, X. R. and Kirubarajan, T.
Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software,
New York: Wiley, 2001.
- [3] Bar-Shalom, Y.
Update with out-of-sequence measurements in tracking: Exact Solution,
IEEE Transactions on Aerospace and Electronic Systems, 38, 3 (Jul. 2002), 769–778.
- [4] Bar-Shalom, Y., Chen, H. M., and Mallick, M.
One-Step Solution for the Multistep Out-of-Sequence Measurement Problem in Tracking,
IEEE Transactions on Aerospace and Electronic Systems, 40, 1 (Jan. 2004), 27–37.
- [5] Bar-Shalom, Y., Willett, P. K. and Tian, X.
Tracking and Data Fusion: A Handbook of Algorithms,
YBS Publishing, 2011.
- [6] Le Cadre, J. P. and Jauffret, C.
Discrete-time observability and estimability analysis for bearings-only target motion analysis,
IEEE Transactions on Aerospace and Electronic Systems, 33, 1 (Jan. 1997), 178–201.
- [7] Clavard, J., Pillon, D., Pignol, A-C. and Jauffret, C.
Target motion analysis of a source in a constant turn from a nonmaneuvering observer,
IEEE Transactions on Aerospace and Electronic Systems, 49, 3 (Jul. 2013), 1760–1780.
- [8] Deb, S., Yeddanapudi, M., Pattipati, K. R. and Bar-Shalom, Y.
A generalized S-D assignment algorithm for multisensor-multitarget state estimation,
IEEE Transactions on Aerospace and Electronic Systems, 33, 2 (Apr. 1997), 523–538.
- [9] Hilton, R. D., Martin, D. A., and Blair, W. D.
“Tracking with time-delayed data in multisensor systems,”
NSWCDD/TR-93/351, Dahlgren, VA, Aug. 1993.
- [10] Jauffret C. and Bar-Shalom, Y.
Track Formation with Bearing and Frequency Measurements in Clutter,
IEEE Transactions on Aerospace Electronic Systems, 26, 6 (Nov. 1990), 999–1009.
- [11] Jauffret, C. and Pillon, D.
Observability in passive target motion analysis,
IEEE Transactions on Aerospace and Electronic Systems, 32, 4 (Oct.1996), 1290–1300.
- [12] Jauffret, C.
Observability and Fisher information matrix in nonlinear regression,
IEEE Transactions on Aerospace and Electronic Systems, 43, 2 (Apr. 2007), 756–759.
- [13] Jauffret, C., Pillon, D., and Pignol, A-C.
Bearings-only maneuvering target motion analysis from a nonmaneuvering platform,
IEEE Transactions on Aerospace and Electronic Systems, 46, 4 (Oct. 2010), 1934–1948.
- [14] Jauffret, C. and Pignol, A-C.
Target motion analysis by inverse triangulation,
Proc. 18th International Conference on Information Fusion,
Washington, DC, Jul. 2015.
- [15] Julier, S. J., and Uhlmann, J. K.
A new extension of the Kalman filter to nonlinear systems,
Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls, Apr. 1997.
- [16] Lindgren, A. G., and Gong, K. F.
Position and velocity estimation via bearing observations,
IEEE Transactions on Aerospace and Electronic Systems, 14, 4 (Jul. 1978) 564–577.
- [17] Nardone, S. C., and Aidala, V. J.
Observability criteria for bearings-only target motion analysis,
IEEE Transactions on Aerospace and Electronic Systems, 17, 2 (Mar. 1981), 162–166.
- [18] Passerieux, J. M., Pillon, D., Blanc-Benon, P., and Jauffret, C.
Target Motion Analysis with Bearing and Frequencies Measurements,
Proceedings of the 22nd Asilomar Conference, Pacific Grove, CA, USA, Nov. 1988.
- [19] Pattipati, K. R., Deb, S., and Bar-Shalom, Y.
A new relaxation algorithm and passive sensor data association,
IEEE Transactions on Automatic Control, 37, 2 (Feb. 1992), 198–213.
- [20] Ristic, B., Arulampalam, S., and Gordon N.
Beyond the Kalman Filter: Particle Filters for Tracking Applications,
Boston London: Artech House, 2004.
- [21] Tichavsky, P., Muravchik, C. H., and Nehorai A.
Posterior Cramer-Rao bounds for discrete-time nonlinear filtering,
IEEE Transactions on Signal Process, 46, 5 (May 1998), 1386–1396.

- [22] Yang, R., Bar-Shalom, Y., and Ng, G. W. Tracking/fusion and deghosting with Doppler frequency from two passive acoustic sensor, *Proc. 16th International Conference on Information Fusion*, Istanbul, Turkey, Jul. 2013.
- [23] Yang, R., Bar-Shalom, Y., Huang, J. H., and Ng, G. W. Interacting multiple model unscented Gauss-Helmert filter for bearings-only tracking with state-dependent propagation delay, *Proc. 17th International Conference on Information Fusion*, Salamanca, Spain, Jul. 2014.
- [24] Yang, R., Bar-Shalom, Y., Huang, J. H., and Ng, G. W. UGHF for acoustic tracking with state-dependent propagation delay, *IEEE Transactions on Aerospace and Electronic Systems*, 51, 3 (Jul. 2015), 1747–1761.
- [25] Yang, R., Bar-Shalom, Y., and Ng, G. W. Bearings-only tracking with fusion from heterogenous passive sensors: ESM/E0 and acoustic, *Proc. 18th International Conference on Information Fusion*, Washington, DC, Jul. 2015.



Rong Yang received her B.E. degree in information and control from Xi’an Jiao Tong University, China in 1986, M.Sc. degree in electrical engineering from the National University of Singapore in 2000, and Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore in 2012. She is currently a Principal Member of Technical Staff at DSO National Laboratories, Singapore. Her research interests include passive tracking, low observable target tracking, GMTI tracking, hybrid dynamic estimation and data fusion. She received the FUSION 2014 Best Paper Award (First runner up).

Yaakov Bar-Shalom was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in electrical engineering. From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, California. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. He is also Director of the ESP (Estimation and Signal Processing) Lab. His current research interests are in estimation theory, target tracking and data fusion. He has published over 500 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate texts *Estimation and Tracking: Principles, Techniques and Software* (Artech House, 1993; translated into Russian, MGTU Bauman, Moscow, Russia, 2011), *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction* (Wiley, 2001), the advanced graduate texts *Multitarget-Multisensor Tracking: Principles and Techniques* (YBS Publishing, 1995), *Tracking and Data Fusion* (YBS Publishing, 2011), and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Artech House, Vol. I, 1990; Vol. II, 1992; Vol. III, 2000). He has been elected Fellow of IEEE for “contributions to the theory of stochastic systems and of multi-target tracking.” He has been consulting to numerous companies and government agencies, and originated the series of Multitarget-Multisensor Tracking short courses offered via UCLA Extension, at Government Laboratories, private companies and overseas. During 1976 and 1977 he served as Associate Editor of the IEEE Transactions on Automatic Control and from 1978 to 1981 as Associate Editor of Automatica. He was Program Chairman of the 1982 American Control Conference, General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–87 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–89 was a member of the Board of Governors of the IEEE CSS. He was a member of the Board of Directors of the International Society of Information Fusion (1999–2004) and served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications in 2004–13. In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS and has given numerous keynote addresses at major national and international conferences. He is co-recipient of the M. Barry Carlton Award for the best paper in the IEEE Transactions on Aerospace and Electronic Systems in 1995 and 2000 and recipient of the 1998 University of Connecticut AAUP Excellence Award for Research. 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. In 2015, he received from the International Society for Information Fusion the Lifetime of Excellence in Information Fusion award. 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.





Gee Wah Ng received his M.Sc. and Ph.D. from University of Manchester Institute of Science and Technology, United Kingdom. He is currently a Distinguished Member of Technical Staff and Programme Director (Information Exploitation) of Information Division at DSO National Laboratories. He has delivered many projects in the decision support areas and has authored three books. He is active in international conferences in the areas of information fusion and intelligent systems. His research interests in data and information fusion include target tracking, computational intelligence, machine learning, self-tuning and sensor networks.

Joint Identification of Multiple Tracked Targets

DOMINIC E. SCHAUB

This paper derives a rigorously Bayesian technique for estimating the identities of a plurality of targets that are well separated or tracked using the (joint) probabilistic data association filter. In contrast to the single-target classification problem, the joint identification of multiple targets is characterized by statistical dependencies between track-to-identity assignments that render track-level estimation of identity suboptimal. The present method rigorously accounts for these dependencies and allows arbitrary feature and kinematic measurements generated by individual targets to be used in finding the statistically-optimal track-to-identity assignment probabilities. The problem is decomposed into global combinatorial identity deconfliction and local target tracking and classification that is based on a unified measure-theoretic filtering framework. The computational complexity of this technique is shown to be dominated by calculation of the permanent of a non-negative matrix, which may be found exactly in exponential time or approximated in polynomial time using Markov chain Monte Carlo methods. Strategies for improving numerical performance are given for cases where certain subsets of targets are indistinguishable or unobservable. This work is relevant to applications in tactical settings, surveillance, including video tracking, air/land/maritime situational awareness, and automated intelligence collection.

Manuscript received February 25, 2015; revised February 7, 2016; released for publication June 21, 2016.

Refereeing of this contribution was handled by Dr. Paolo Braca.

Author's address: Defence Research and Development Canada, Atlantic Research Centre, 9 Grove St., P.O. Box 1012, Dartmouth NS B2Y 3Z7 Canada (E-mail: dominic.schaub@drdc-rddc.gc.ca).

This work was supported by DRDC Applied Research Project 06eo, *Situational Information for Enabling Development of Northern Awareness* (SEDNA).

1. INTRODUCTION

Multisensor tracking is a branch of information fusion that appears in many domains that require situational awareness, such as air traffic control, video surveillance, and missile defence. The fundamental task in tracking involves estimating target kinematic states (such as position and/or velocity of an aircraft) from a time series of measurements generated by a suite of one or more sensors (e.g. radar). While several approaches have been developed for fusing measurement data, those based on (or approximating) Bayesian filtering [1]–[3] have enjoyed the widest adoption owing to their rigorous treatment of sensor error and target dynamics. In the Bayesian framework, a probability distribution over target state space is maintained and updated with new measurements using Bayes' theorem. In particular, where sensor noise and target dynamics admit a priori statistical characterizations, Bayesian methods offer a rigorously mathematical framework for computing optimal statistical estimates.

Over the course of several decades, enormous advances in sensors, communications systems, and computing power have enabled the development of a considerable number of tracking methods, from the simple, linear Kalman filter [4], [5] (which solves the single-target state-estimation problem under Gaussian conditions) to the sophisticated multi-hypothesis tracker [6] (which performs the task of associating multiple observations to multiple tracks and is used in conjunction with a collection of individual filters) and the probability hypothesis density (PHD) filter [7], [8] (which tightly integrates data association and filtering). Research has also broadened to include the related problems of target identification and classification, which naturally extend the mathematics of target tracking. In principle, these undertakings may be regarded as specializations of the same fundamental problem, as identification amounts to a constrained form of classification that assigns a given class to at most a single target.

As with tracking, identification and classification are marked by significant differences between their single- and multiple-target specializations. Single-target joint tracking and classification (JTC) has been extensively studied as a rigorously Bayesian problem, with detailed theoretical derivations provided in [9]–[11]. Particular attention has been given to exploiting kinematic data to assist classification, including [12], which demonstrated improvements in classification performance by applying a second-order uncertainty model to the mapping between the feature and target class spaces. Similarly, [13] developed a framework of multiple-model particle and mixture Kalman filters subject to kinematic constraints and subsequently considered its application to discriminating between commercial and military aircraft

using radar contacts. In [14], it was shown that maximum entropy techniques can significantly improve accuracy in Bayesian classification characterized by epistemic uncertainty in the prior (which, when unknown, is typically taken as uniform). Specific applications have also been considered, including a decision-theoretical problem of identifying aircraft using radar measurements [15] and a joint tracking and classification framework for radar and electronic-support-measure observations [16]. It is worth highlighting that Bayesian single-target joint tracking and classification is typically computationally feasible for modest-dimensional problems, where the ‘curse of dimensionality’ (in the non-parametric case) is usually mild and may be overcome through particle filtering or carefully implemented fixed-grid discretizations.

In the multitarget context, identification has been theoretically analyzed as an extension to Finite Set Statistics (FISST) using the framework of labelled random sets [17]. Several approximation schemes have also been developed (albeit lacking in rigorous tracking formalism), including methods based on information theory [18], [19] and Sinkhorn rescaling [20]–[22]. Accounts of relating specific tracking implementations with the higher-level identification problem have been given in [23], which developed (using a series of approximations) a multitarget Kalman filter that utilizes target identity information. In particular, a group-theoretical Fourier method [24]–[26] has emerged as a general framework for approximate reasoning over combinatorial matchings. In this approach, distributions defined over the set of permutations are replaced with equivalent (Fourier-transformed) distributions over the irreducible representations of the related symmetric group. Under favourable conditions, the transformed quantities may be approximated with a small number of terms, thereby avoiding the factorial space and time complexities ordinarily encountered in combinatorial problems.

The foregoing methods of multitarget identification are either intractable (implementation of the random-set formulation entails further numerical simplifications) or dependent on complex approximations that impede the analysis of error in computed estimates. These limitations are a consequence not only of the inherent complexity of multitarget tracking, but of the fact that multitarget identification and constrained classification—where at most a fixed number of targets may belong to a given class¹—, is itself non-local and combinatorial, as estimating the class of a target depends partly on observations made at distant tracks (e.g. a target at a given track is unlikely to be a particular identity when there is strong evidence for its presence elsewhere). Interestingly, the underlying mathematical structure—known as

¹For example, a given task group may include a known number of UAVs and helicopters. In this case, the fixed number of each aircraft imposes a global constraint on the multitarget classification problem.

the assignment problem [27], [28]—is of considerable generality, appearing in several diverse contexts such as economics [29], [30], operations research [31], [32], and the joint probabilistic data association (JPDA) filter [33].

The present work develops a special case of rigorously Bayesian multitarget identification and classification wherein it becomes computationally feasible to calculate nearly exact estimates of identity and kinematic states. The necessary conditions are met whenever measurement-to-track associations are unambiguous, i.e., where tracks are well separated or, alternatively, where (J)PDA is used to resolve the data association problem. The primary objectives motivating this work lie with computing the marginal track-to-identity assignments in the form of posterior probabilities that a given identity—such as a vessel with a particular registration number—is present at a track of interest (Fig. 1) and, in addition, finding the optimal posterior target-state-space densities from which kinematic estimates may be calculated. It is shown that the algorithmic bottleneck stems from calculation of the matrix permanent, a standard function in combinatorics and one that may be efficiently computed using rapidly-mixing Markov chain Monte Carlo (MCMC) methods [34]–[36]. As these techniques yield approximations whose residual errors decrease exponentially with iteration number, optimal Bayesian estimates of any quantity may therefore be found to machine precision in polynomial time. It is further shown that the Ryser method [37], [38] (an analytical permanent algorithm) can be extended to exploit the mathematical structure present when targets form classes of indistinguishable members, a result that enables the *exact* constrained classification of very large numbers of targets to be performed efficiently.

The remainder of this paper is organized as follows. Section 2-A derives a general Bayesian framework that unifies the problems of tracking, classification, and identification under a single statistical system that fully accounts for their complex mathematical interdependence and makes optimal use of arbitrary feature and kinematic observations originating from individual tracks. In §2-B, the framework is equipped with simplifying conditions, including the requirement for unambiguous measurement-to-track associations, which in turn give way to an efficient factorization of the joint target density. This factorization naturally partitions the framework into the connected problems of local tracking and classification and global combinatorial deconfliction of identity, thereby significantly improving the associated time and space complexities. Section 2-C proceeds to show that the central task of deconfliction amounts to computing the permanent of a non-negative matrix, and in §2-D, the framework is extended to applications where data association is performed with (J)PDA. Section 3 then discusses algorithms for computing the matrix permanent, including a Ryser method that is modified to exploit the presence of unobservable

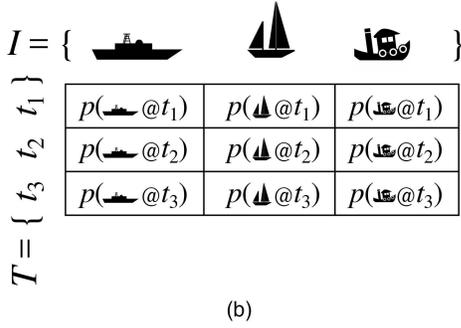
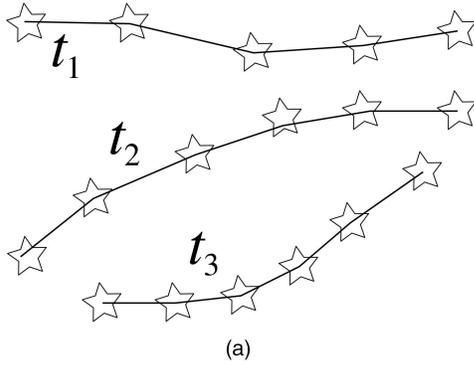


Fig. 1. Simplified overview of the problem studied in this paper. Given a set T of tracks with associated observations (a) and a second set I of unique identities, it is desired to find the matrix of posterior marginal track-to-identity assignment probabilities (b).

or indistinguishable targets to improve runtime (§3-A and §3-B) and various approximation methods, including Markov chain Monte Carlo techniques and loopy belief propagation (§3-C). Finally, a series of numerical examples are considered in §4, demonstrating how this work may be applied to benefit situational awareness.

2. BAYESIAN FORMULATION

A. Overview

This section derives a measure-theoretic Bayesian framework for multitarget filtering and classification in cases where target dynamics and measurements satisfy standard Markov conditions. Development proceeds by augmenting the conventional analytical representation of Bayesian multitarget tracking [8] with a singular-measure extension that allows seamless integration of classification based on static attributes. In what follows, an ‘identity’ refers to a physical entity of uncertain location that possesses a known signature, which can be represented by a combination of dynamic and static features.² Consistent with conventional tracking nomenclature, ‘target’ denotes the entity present at a given track, which is localized to a particular region in space but is of

²For example, each registered vessel on a lake would constitute a unique identity. Note that for a given suite of sensors, identities may not necessarily be distinguishable by way of their observed signatures, as occurs when identical models of watercraft are tracked with radar. In practical applications, the set of identities may be realized as a comprehensive database of physical assets (and their signatures) that could be encountered by the tracking system.

uncertain identity. The sets of all tracks and identities in existence are designated T and I , respectively (Fig. 1). Furthermore, these sets are well ordered, allowing each of their members to be referenced by a unique integer index.

The present framework is simplified when each identity is assignable to a distinct track. To accommodate circumstances where observed tracks are produced by a series of false alarms (or where there are fewer observed tracks than identities), it may be mathematically convenient to enlarge T by including additional tracks whose targets are never observed. Without loss of generality, the total set of tracks may thus be defined by the union

$$T = T_O \cup T_H \quad (1)$$

where T_O are those tracks with observed targets (to which one or more measurements are associated), and T_H is a (possibly empty) set of hidden tracks lacking measurements. The set T_H can be made identical in size to I to allow for the possibility that every track in T_O was generated by false alarms.

Each identity $i \in I$ assumes values in its associated state space (\mathbb{S}_i). For mathematical simplicity, an arbitrary identity state $\mathbf{x}_i \in \mathbb{S}_i$ is made to encapsulate both dynamic attributes (e.g. velocity) and static properties (e.g. length) in a manner that allows the state space to be decomposed as the Cartesian product

$$\mathbb{S}_i = \mathbb{S}_{i,d} \times \mathbb{S}_{i,s} \quad (2)$$

where $\mathbb{S}_{i,d}$ and $\mathbb{S}_{i,s}$ are the dynamic and static components, respectively. Both $\mathbb{S}_{i,d}$ and $\mathbb{S}_{i,s}$ must be equipped with σ -algebras, allowing \mathbb{S}_i to be assigned the measure

$$\mu_i(W) = \lambda_i(\text{pr}_d W) \times \delta_i(\text{pr}_s W) \quad (3)$$

where $W \subseteq \mathbb{S}_i$ is an arbitrary measurable set, $\text{pr}_{(\cdot)}$ are projection operators, and $\lambda_i(\cdot)$ and $\delta_i(\cdot)$ are measures on $\mathbb{S}_{i,d}$ and $\mathbb{S}_{i,s}$, respectively. The latter is the (singular) Dirac measure that encodes the given identity’s static attribute information. For example, where there exist two identities of length 1 m and 2 m, their respective Dirac measures on their ‘length’ spaces would be $\delta_{1\text{ m}}(\cdot)$ and $\delta_{2\text{ m}}(\cdot)$. For an observation of target length given by³ $\mathcal{N}(\mu = 1\text{ m}, \sigma^2 = 1\text{ m}^2)$, the likelihoods associated with each of the identities may be found by applying their Dirac measures to the measurement’s pdf, and are thus $(2\pi)^{-1/2}$ and $e^{-1}(2\pi)^{-1/2}$ for the 1-m and 2-m identities, respectively. However, rather than extracting feature information directly from individual observations, it may be convenient to maintain a running product of static-feature measurements from previous updates—these products are simply individual measurements pdfs (pmfs) that are combined by successive applications of

³Obviously, the normal distribution implies that lengths could assume negative (non-physical) values.

Bayes' rule.⁴ Invocation of the identity-specific Dirac measures may then be deferred to the time at which the posterior probabilities are computed. By transferring the classification operations to the measure, static features are placed on equal footing as the dynamic components, allowing Bayesian feature measurement updates to be performed in manner identical to their kinematic counterparts. Note that this formulation simplifies the subsequent presentation but is theoretically equivalent to that derived with intermediate feature spaces.

Multitarget tracking and classification may be formulated with Cartesian products of individual identity spaces that are constructed as

$$\mathbb{S}_J = \prod_{i \in J} \mathbb{S}_i, \quad J \subseteq I, \quad (4)$$

where the product order is given by the total order on I , and the subscripts are suppressed when $J = I$. The \mathbb{S}_J of (4) are similarly assigned the family of product measures

$$\mu_J(W) = \prod_{i \in J} \mu_i(\text{pr}_i W). \quad (5)$$

Uncertainty in the system's dynamic state may be represented with a random variable X on \mathbb{S} whose probability density function $p_X(\mathbf{x})$ is uniquely defined up to an equivalence class determined by the measures of (3). In particular, each equivalence class contains a member that assumes strictly constant values on each of the static attribute spaces and thus satisfies

$$p_{X_i}(\mathbf{x}_i) = p_{X_i}(\mathbf{x}'_i), \quad \forall \mathbf{x}_i, \mathbf{x}'_i \in \mathbb{S} : \text{pr}_d\{\mathbf{x}_i\} = \text{pr}_d\{\mathbf{x}'_i\} \quad (6)$$

which uses the fact that all functions on the i th static attribute space that evaluate to the same quantity for the argument selected by $\delta_i(\cdot)$ are equivalent under the associated measure (e.g. $f_1(x) = 1/\sqrt{2}$, $f_2(x) = |\sin(x)|$, and $f_3(x) = |\cos(x)|$ are equivalent under $\delta_{\pi/4}(\cdot)$). Priors on static attribute spaces may therefore be specified as constant functions. Marginalization of this density with respect to $J \subset I$ may be defined as

$$p_{X_{I \setminus J}}(\mathbf{x}_{I \setminus J}) = \int_{\mathbb{S}_J} p_X(\mathbf{x}) d\mu_J(\mathbf{x}_J). \quad (7)$$

The system state is updated with a sequence of discrete-time measurements \mathfrak{z}^k , which are samples of

the time series of random variables $Z^k \in \mathbb{S}_Z$ that describe the measurement processes. The collection of measurements up to and including the k th time step is given by

$$\mathfrak{z}^{1:k} = \{\mathfrak{z}^{k'}\}_{k'=1}^k. \quad (8)$$

For each time step there exists a likelihood function $f : (\mathbb{S}_Z, \mathbb{S}) \rightarrow [0, \infty)$ that carries information on the measured static and/or dynamic attributes. When conditioned on the multitarget state and evaluated for a given measurement $Z^k = \mathfrak{z}^k$, this function may be written as

$$f_{Z^k|X^k}(\mathfrak{z}^k | \mathbf{x}). \quad (9)$$

Between measurements, the dynamic components of the state are projected forward in time by a Markov transition density $f : (\mathbb{S}, \mathbb{S}) \rightarrow [0, \infty)$ that encodes the known aspects of the identities' dynamics (e.g. acceleration, etc.). It is similarly conditioned on the k th state and is denoted by

$$f_{X^{k+1}|X^k}(\mathbf{x} | \mathbf{x}'). \quad (10)$$

Bayesian filtering consists of alternately updating the state with new measurements using Bayes' rule (usually increasing the Fisher information) and transitioning the state forward in time (usually decreasing the information). The measurement and transition steps may be written as

$$p_{X^k|Z^{1:k}}(\mathbf{x} | \mathfrak{z}^{1:k}) = \frac{f_{Z^k|X^k}(\mathfrak{z}^k | \mathbf{x}) p_{X^k|Z^{1:k-1}}(\mathbf{x} | \mathfrak{z}^{1:k-1})}{\int_{\mathbb{S}} f_{Z^k|X^k}(\mathfrak{z}^k | \mathbf{x}') p_{X^k|Z^{1:k-1}}(\mathbf{x}' | \mathfrak{z}^{1:k-1}) d\mu(\mathbf{x}')} \quad (\text{BF.1})$$

and

$$p_{X^{k+1}|Z^{1:k}}(\mathbf{x} | \mathfrak{z}^{1:k}) = \int_{\mathbb{S}} f_{X^{k+1}|X^k}(\mathbf{x} | \mathbf{x}') p_{X^k|Z^{1:k}}(\mathbf{x}' | \mathfrak{z}^{1:k}) d\mu(\mathbf{x}') \quad (\text{BF.2})$$

respectively. By virtue of the product measures of (3), these equations simultaneously subsume both tracking and classification (two specializations that occur in the absence of $\mathbb{S}_{s,i}$ and $\mathbb{S}_{d,i}$, respectively).

B. Tracking and Identification of Well-Separated Targets

The general framework of the previous section is challenging to implement numerically. For example, a fixed-grid discretization of a probability density function exhibits time and space (storage) complexities⁵ that scale exponentially with the dimension of \mathbb{S} (which is proportional to cardinality of I), precluding even approximate evaluation in all but the most elementary cases. The present work exploits the fact that in many instances, targets are far apart and non interacting. In

⁴Philosophically, this approach treats fixed features (such as length) as time-indexed probability distributions (akin to the distributions describing kinematics) subject to standard Bayesian recursive filtering. By choosing the prior feature density to be the Dirac delta function (or Kronecker delta function where the prior is a pmf) and defining the associated Markov transitions to have no effect, the feature posterior will remain the Dirac delta function indefinitely, as no measurement function can increase its information further. However, the delta function's magnitude will be scaled in accordance with how 'close' the feature measurements are to the feature encoded in the delta function. The Dirac measure is substituted for the Dirac delta function to avoid the mathematical sophistication required to use the latter.

⁵An overview of computational complexity theory may be found in [39].

this case, measurement-to-track associations are unambiguous, and the problem inherits several simplifying characteristics ordinarily associated with single-target tracking. Those simplifications form the basis the remaining analysis and are described below.

- 1) Individual identity state spaces are derived from a single common space. That is,

$$\mathbb{S}_i = \mathcal{S} \quad \forall i \in I. \quad (11)$$

In practice, a sufficiently large \mathcal{S} can be defined to encompass any set of original identity state spaces.

- 2) To enforce a clear separation of targets, the probability density functions maintained at each track⁶ are confined to a unique set $r_t^k \subset \mathcal{S}$ indexed by track and time. These target regions need not be explicitly defined, provided that a disjoint construction ($r_t^k \cap r_{t'}^k = \emptyset$ for all $t \neq t'$) is guaranteed to exist—an assumption that is reasonable when targets are well separated. For example, where the t th target is tracked in two dimensions using Kalman filtering, the positional component of r_t^k may be defined as the area circumscribed by a few standard deviations from the means of the normally-distributed posteriors.⁷ In what follows, the collection of all disjoint regions at the k th time step is given as

$$R^k = \{r_t^k\}_{t \in T}, \quad (12)$$

from which it may be seen that each r_t^k corresponds to a distinct t , and $|R^k| = |T|$. As each identity may be found in one of the regions, the following holds

$$\Pr \left\{ X_i^k \in \bigcup_{t \in T} r_t^k \right\} = 1, \quad \forall i \in I, \quad (13)$$

where X_i^k (shorthand for $X_{\{i\}}^k$) is the i th-identity marginalization of X^k . In the present work, each track (region) is restricted to contain at most one target, yielding

$$\begin{aligned} 0 &\leq \sum_{\substack{J \subseteq I \\ |J| \text{ fixed}}} \Pr \{X_J^k \in (r_t^k)^{|J|} \wedge X_{I \setminus J}^k \in \mathbf{C}(r_t^k)^{|I \setminus J|}\} \\ &\leq \begin{cases} 1 & |J| = 1 \\ 0 & |J| > 1 \end{cases} \quad \forall r_t^k \in R^k, \end{aligned} \quad (14)$$

where \mathbf{C} , $|\cdot|$, and $(r_t^k)^{|\cdot|}$ denote set complementation, cardinality, and Cartesian products of r_t^k , respectively. As an immediate consequence of (13) and (14), each i -indexed component of any random sample of X^k must always fall upon a distinct r_t^k . Thus, as foreshadowed in §2-A, the number of disjoint re-

gions (and hence the number of tracks) must match or exceed the number of identities. When the rightmost inequality in (14) is strict, the affected region may not actually contain a target, and the probability of existence of the corresponding track is therefore less than one.

The restrictions imposed by (13) and (14) may be expressed in terms of a time-indexed union of admissible subsets A_σ^k of the multitarget state space \mathbb{S} . Each subset corresponds to a single injective function σ between I and T and is defined as the Cartesian product

$$A_\sigma^k = \prod_{i \in I} r_{\sigma(i)}^k. \quad (15)$$

The collection of σ forms the set of all identity-track permutations $P(I, T)$. Thus, the admissible subset of $A^k \subset \mathbb{S}$ is given by the union

$$A^k = \bigcup_{\sigma \in P(I, T)} A_\sigma^k. \quad (16)$$

In turn, the A_σ^k may be used to construct the indicator functions $\mathbf{1}_{A_\sigma^k}(\mathbf{x})$ and $\mathbf{1}_{A^k}(\mathbf{x})$ defined on \mathbb{S} . Moreover, every $\mathbf{1}_{A_\sigma^k}(\mathbf{x})$ can be factored into identity-specific components as

$$\mathbf{1}_{A_\sigma^k}(\mathbf{x}) = \prod_{i \in I} \mathbf{1}_{r_{\sigma(i)}^k}(\mathbf{x}_i). \quad (17)$$

By virtue of the fact that $A_{\sigma_1}^k \cap A_{\sigma_2}^k = \emptyset$ for $\sigma_1 \neq \sigma_2$, the indicator functions also satisfy

$$\mathbf{1}_{A^k}(\mathbf{x}) = \sum_{\sigma \in P(I, T)} \mathbf{1}_{A_\sigma^k}(\mathbf{x}). \quad (18)$$

- 3) The joint density function admits a quasi statistically independent factorization as the product of a scaled indicator function and a set of normalized i -indexed probability density functions $p_i^k(\mathbf{x}_i)$ on \mathcal{S} as

$$p_{X^k}(\mathbf{x}) = C^k \mathbf{1}_{A^k}(\mathbf{x}) \prod_{i \in I} p_i^k(\mathbf{x}_i) \quad (19)$$

where $\mathbf{1}_{A^k}(\mathbf{x})$ zeroes regions of the joint probability density that violate (14), and C^k is a positive constant that renormalizes the product, which is partially zeroed by the indicator function. Note that $p_i^k(\mathbf{x}_i)$ is distinct from $p_{X_i^k}(\mathbf{x}_i)$, which is the $I \setminus \{i\}$ -marginalization found with (7). The $p_i(\mathbf{x}_i)$ may themselves be decomposed as sums of region (track)-specific normalized density functions to yield

$$\begin{aligned} p_{X^k}(\mathbf{x}) &= C^k D^k \mathbf{1}_{A^k}(\mathbf{x}) \prod_{i \in I} \sum_{t \in T} (b_t^k)^{-1} a_{t,i}^k p_{t,i}^k(\mathbf{x}_i) \\ \text{supp}(p_{t,i}^k) &\subseteq r_t^k \end{aligned} \quad (20)$$

where $\text{supp}(\cdot)$ denotes function support, and the $a_{t,i}^k$, b_t^k , and D^k are non-negative coefficients that are consistent with C^k and $p_i^k(\mathbf{x}_i)$ in (19). To ensure the existence of $(b_t^k)^{-1}$, the b_t^k are further required to be positive. The C^k , D^k , and b_t^k are extraneous (they may be subsumed in the $a_{t,i}^k$) but simplify subsequent

⁶These are explicitly defined in condition 3. For the moment, they may be regarded as probability density functions produced by single-target trackers.

⁷In actual fact, the normal distribution has unbounded support, and the separation condition therefore fails to hold. This may be overcome by treating the Kalman filter as an approximation that operates on truncated Gaussian functions.

analysis. Also note the relation

$$\mathbf{1}_{r_i^k}(\mathbf{x}_t) p_{t',i}^k(\mathbf{x}_t) = \begin{cases} p_{t',i}^k(\mathbf{x}_t), & t = t' \\ 0, & t \neq t'. \end{cases} \quad (21)$$

The factorizations of (19) and (20) allow the full joint distribution to be efficiently represented by series of constants that grows only linearly with the number identities and tracks, thereby immensely reducing the problem's computational complexity.

- 4) Target motion is assumed independent, allowing the Markov transition densities to be factored as

$$\begin{aligned} f_{X^{k+1}|X^k}(\mathbf{x} | \mathbf{x}') &= \prod_{i \in I} f_{X_i^{k+1}|X_i^k}(\mathbf{x}_i | \mathbf{x}'_i) \\ &= \prod_{i \in I} \sum_{t \in T} f_{t, X_i^{k+1}|X_i^k}(\mathbf{x}_i | \mathbf{x}'_i) \end{aligned} \quad (22)$$

where $\text{supp}(f_{t, X_i^{k+1}|X_i^k})$ is a subset of the Cartesian product $r_i^{k+1} \times r_i^k$.

- 5) Observations are assumed to be of type produced in single-target Bayesian filtering. Each successive time step therefore corresponds to only a single observation at one track, where the history of observation-to-track associations is given by the time-indexed observation vector $\mathbf{o}(k)$ that maps each time step k to a single track t . To update the multitarget prior, a global measurement likelihood function must be constructed from a single-target, single-track observation function $f_{Z^k|X_S^k}(\mathfrak{Z}^k | \mathbf{x} \in \mathcal{S})$ as follows

$$\begin{aligned} f_{Z^k|X^k}(\mathfrak{Z}^k | \mathbf{x}) &= \mathbf{1}_{A^k}(\mathbf{x}) \left(\sum_{i \in I} p_{d,i}^k f_{Z^k|X_S^k}(\mathfrak{Z}^k | \mathbf{x}_i) + p_f^k \right), \\ \text{supp}(f_{Z^k|X_S^k}) &\subseteq r_{\mathbf{o}(k)}, \end{aligned} \quad (23)$$

where $t = \mathbf{o}(k)$ is the track that produced the observation, $p_f^k \geq 0$ is the sensor false-alarm probability, $p_{d,i}^k \geq 0$ is the identity detection probability, and the leading indicator term is included for mathematical convenience (it is idempotent, and owing to its appearance in the prior, has no effect when used in the Bayesian update). Expressed as a sum, the measurement function does not obviously preserve the quasi statistical independence in (19). However, when $p_f^k > 0$, the properties of the indicator function may be used to rewrite (23) as

$$\begin{aligned} f_{Z^k|X^k}(\mathfrak{Z}^k | \mathbf{x}) &= \\ & (p_f^k)^{1-|I|} \mathbf{1}_{A^k}(\mathbf{x}) \prod_{i \in I} (p_{d,i}^k f_{Z^k|X_S^k}(\mathfrak{Z}^k | \mathbf{x}_i) + p_f^k), \end{aligned} \quad (24)$$

which is easily shown to leave the structure of (19) intact. While this reformulation introduces an $(|I| - 1)$ th-order singularity with respect to p_f^k , it is removable in the regions of \mathbb{S} that are not zeroed by the indicator function, as it cancels with a powers of p_f^k greater than or equal to $|I| - 1$. Thus, while (24)

is undefined for $p_f^k = 0$, the right-hand sides of (23) and (24) share the same pointwise limit $p_f^k \rightarrow 0^+$, a property that may be used in cases where false-alarm probabilities are zero (a formal limiting procedure is derived in Appendix B). Finally, (24) may be algebraically recast as

$$\begin{aligned} f_{Z^k|X^k}(\mathfrak{Z}^k | \mathbf{x}) &= p_f^k \mathbf{1}_{A^k}(\mathbf{x}) \\ & \cdot \prod_{i \in I} (\mathbf{1}_{r_i^k}(\mathbf{x}_i) (p_f^k)^{-1} g_{i, \mathfrak{Z}^k}(\mathbf{x}_i) + \mathbf{1}_{\mathbb{C}_{r_i^k}}(\mathbf{x}_i)), \end{aligned} \quad (25)$$

where

$$g_{i, \mathfrak{Z}^k}(\mathbf{x}_i) = p_{d,i}^k f_{Z^k|X_S^k}(\mathfrak{Z}^k | \mathbf{x}_i) + p_f^k. \quad (26)$$

By induction, conditions 4–5 may be proved consistent with condition 3 for all time instants, provided that (19) holds at $k = 0$. Of particular significance is the fact that, while the X_i^k are not statistically independent, their joint probability density function $p_{X^k}(\mathbf{x})$ nonetheless admits an efficient factorization that is readily updated with new information from local tracks. The Bayesian filtering problem therefore amounts to iteratively computing $a_{t,i}^k$, b_t^k , D^k , and $p_{t,i}^k(\mathbf{x}_i)$ from previous values. In accordance with the derivation in Appendix A, the k th time-step quantities may be related to those at $k - 1$ by

$$\begin{aligned} p_{t,i}^k(\mathbf{x}_i) &= (K_i^k)^{-1} g_{i, \mathfrak{Z}^k}(\mathbf{x}_i) \\ & \cdot \int_{r_i^k} f_{X_i^k|X_i^{k-1}}(\mathbf{x}_i | \mathbf{x}'_i) p_{t,i}^{k-1}(\mathbf{x}'_i) d\mu(\mathbf{x}'_i) \\ a_{t,i}^k &= K_i^k a_{t,i}^{k-1} \\ b_t^k &= p_t^k b_t^{k-1} \end{aligned} \quad (27)$$

for $t = \mathbf{o}(k)$, and

$$\begin{aligned} p_{t,i}^k(\mathbf{x}_i) &= \int_{r_i^k} f_{X_i^k|X_i^{k-1}}(\mathbf{x}_i | \mathbf{x}'_i) p_{t,i}^{k-1}(\mathbf{x}'_i) d\mu(\mathbf{x}'_i) \\ a_{t,i}^k &= a_{t,i}^{k-1} \\ b_t^k &= b_t^{k-1} \end{aligned} \quad (28)$$

for $t \neq \mathbf{o}(k)$, with K_i and D^k given by

$$\begin{aligned} K_i^k &= \int_{r_i^k \times r_i^{k-1}} g_{i, \mathfrak{Z}^k}(\mathbf{x}_i'') f_{X_i^k|X_i^{k-1}}(\mathbf{x}_i'' | \mathbf{x}_i') \\ & \cdot p_{t,i}^{k-1}(\mathbf{x}_i') d\mu(\mathbf{x}_i'') d\mu(\mathbf{x}_i') \\ D^k &= p_f^k D^{k-1} = \prod_{i \in T} b_i^k. \end{aligned} \quad (29)$$

Equations 27 and 28 illustrate that updating the coefficients may be carried out on a per-track basis. Thus, the problem may be formulated as a collection of individual single-target tracking problems whose identity information exhibit inter-track statistical dependence. Each track t is associated with a single region r_t^k , and a separate Bayesian filter is run for each observed track/identity pair for a total of $|T_0| \cdot |I|$ individual filters.

For a given track, this entails selecting a prior comprising $a_{t,i}^1$, b_t^1 , D^1 , and $p_{t,i}^1(\mathbf{x})$ and then recursively computing the filtering equations of (27) and (28) for each $i \in I$. Note that calculation of the normalization constant C^k is not required at each time step and may therefore be deferred to the final evaluation of the posterior.

Where observations occur simultaneously, measurements may be assigned consecutive time indexes by defining the zero-time-difference Markov transitions as identity maps. Alternatively, simultaneous measurements may be folded into a single time step. When concurrent observations originate from the same track, a composite function may be constructed as the product of individual measurement functions. In the general case—where observations originate from multiple tracks— $\mathbf{o}(k)$ becomes multivalued, and \mathfrak{Z}^k , $p_{d,i}^k$, p_t^k , and K_i^k must be relabeled as \mathfrak{Z}_t^k , $p_{d,t,i}^k$, $p_{t,i}^k$, and $K_{t,i}^k$, respectively.

C. Combinatorial Evaluation of Target Identity (Identity Deconfliction)

Information about the track-to-identity assignment probabilities may be summarized by the matrix \mathbf{P}^k , whose elements $\mathbf{P}_{t,i}^k$ are the marginal probabilities that identity i is located at track t . The k th time-step $\mathbf{P}_{t,i}^k$ are calculated from the $a_{t,i}^k$ and b_t^k coefficients (maintained by the individual trackers), which may be collected in the $|T| \times |T|$ -dimensional matrix⁸

$$\mathbf{M}_{t,j}^k = \begin{cases} a_{t,i}^k, & j \leq |I| \\ b_t^k, & j > |I| \end{cases}, \quad (30)$$

where the possible inclusion and repetition of b_t^k ensures that \mathbf{M}^k is square. In accordance with Appendix B-A, each $\mathbf{P}_{t,i}^k$ may then be recovered from i th-identity marginalization of the multitarget joint density function

$$\begin{aligned} \mathbf{P}_{t,i}^k &= \int_{\mathfrak{X}_i^k} P_{X_i^k|Z^{1:k}}(\mathbf{x}'_i | \mathfrak{Z}^{1:k}) d\mu(\mathbf{x}'_i) \\ &= \mathbf{M}_{t,i}^k \frac{\text{Per}(\mathbf{M}^k(t;i))}{\text{Per}(\mathbf{M}^k)}, \end{aligned} \quad (31)$$

where $\mathbf{M}^k(t,i)$ is the matrix formed by deleting the t th row and i th column from \mathbf{M}^k and $\text{Per}(\cdot)$ denotes the matrix permanent [37], [38] (a combinatorial sum indexed by the $|T|$ th-order symmetric group $S_{|T|}$) defined as

$$\text{Per}(\mathbf{M}^k) = \sum_{\sigma \in S_{|T|}} \prod_{j=1}^{|T|} \mathbf{M}_{\sigma(j),j}^k. \quad (32)$$

Note that the updates to $a_{t,i}^k$, b_t^k , D^k , and $p_{t,i}^k(\mathbf{x})$ are entirely independent of (31), whose evaluation may be deferred to an arbitrary time step, and furthermore, only those $\mathbf{P}_{t,i}^k$ of interest need be computed. As a result of the limiting procedure described in Appendix B-B, the p_t^k

⁸The t and i that index the elements of \mathbf{M}^k are shorthand for the integer indices (over $\{1, \dots, |T|\}$ and $\{1, \dots, |I|\}$) induced by the total orders on T and I , respectively.

associated with zero false-alarm measurements may be set to zero when computing g_{i,\mathfrak{Z}^k} and b_t^k . Consequently, those tracks for which at least one measurement was completely certain (zero false alarm probability) will possess a b_t^k equal to zero.

The integration of identification into tracking also yields improvements to track-level estimates. In particular, the coefficients of \mathbf{P}^k (which contain the marginal track-to-identity assignment probabilities) can be used to scale the individual density functions of (20) in the expansion

$$p_t^k(\mathbf{x}) = \sum_{i \in I} \mathbf{P}_{t,i}^k p_{t,i}^k(\mathbf{x}), \quad (33)$$

where the resulting probability density function $p_t^k(\mathbf{x})$ is the weighted average of the i -indexed posteriors generated by the trackers of the filter bank at track t . Estimation may be performed on $p_t^k(\mathbf{x})$ to compute track-level quantities of dynamic (e.g. kinematic) or static attributes. Interestingly, while the use of multiple models for transition and measurements functions clearly benefits tracking in a range of applications, the global combinatorial deconfliction step (which generates \mathbf{P}^k) improves the performance of local tracking—and any associated estimates—still further by incorporating non-local information into the track-level posterior (e.g. if a unique, fast-moving entity i is observed with high certainty at a remote track $t' \neq t$, the weight $\mathbf{P}_{t,i}^k$ in the sum of (33) will be reduced, and $p_t^k(\mathbf{x})$ will be commensurately improved).

Finally, groups of identities that possess indistinguishable feature measures $\delta(\cdot)$ and Markov transition densities (in (3) and (10), respectively) form equivalence classes that give rise the problem of constrained classification.⁹ The foregoing analytical framework remains unchanged, though it should be noted that assigning a common prior to members of a group of indistinguishable identities will result in identical marginal track-to-identity probabilities ($\mathbf{P}_{t,i}^k$) across members of that class.

D. Tracking and Identification using (J)PDAF

The preceding framework readily extends to circumstances where tracking is performed using the (joint) probabilistic data association filter [40], [41], albeit at the loss of mathematical optimality.¹⁰ In this case, condition 2 of §2-B no longer holds (the marginalized probability density functions of different targets may overlap), and consequently, the joint state cannot be written as (19) or (20), which rely on indicator functions to zero the inadmissible regions of the multitarget space. This may be remedied by rewriting the joint density

⁹Identification is the trivial case that occurs when a given class contains only a single member.

¹⁰Note that while JPDA employs similar combinatorial framework for determining *measurement-to-track associations*, the calculations involved are distinct from those of the identification problem, which seek to find the *track-to-identity assignments*.

function as an explicit summation over identity-track permutations given by

$$p_{\mathcal{X}^k}(\mathbf{x}) = C^k D^k \sum_{\sigma \in P(I,T)} \prod_{i \in I} (b_i^k)^{-1} a_{\sigma(i),i}^k P_{\sigma(i),i}^k(\mathbf{x}_i) \quad (34)$$

With this modification, the track-level processing and global identity deconflation of the previous sections may be retained without further change. However, while the measurement updates may proceed as before, *determination of the updates themselves* (e.g. by nearest neighbour or maximum likelihood in PDA—or combinatorial methods in JPDA) should be performed using the previous timestep’s posterior weights to scale the track-level density functions (i.e., by using $\mathbf{P}_{i,i}^k p_{i,i}^k(\mathbf{x}_i)$ rather than $(b_i^k)^{-1} a_{i,i}^k p_{i,i}^k(\mathbf{x}_i)$). Thus, an optimally-implemented (J)PDA filter requires that the identity deconflation step be carried out prior to each new measurement.

3. EVALUATION OF THE MATRIX PERMANENT

A. Exact Methods

Although the direct evaluation of (32) is readily implemented and numerically stable, its time complexity (for an n -dimensional matrix) of $\mathcal{O}(n \cdot n!)$ is prohibitive in most practical applications. An improvement in running time is realized by evaluating the sum by way of an inclusion-exclusion decomposition

$$\text{Per}(\mathbf{M}^k) = (-1)^n \sum_{Y \in \mathcal{P}(\mathbb{N}_n)} (-1)^{|Y|} \prod_{q=1}^n \sum_{y \in Y} \mathbf{M}_{q,y}^k, \quad (35)$$

where $\mathcal{P}(\mathbb{N}_n)$ is the power set of the first n positive integers. Known as the Ryser formula [37], [38], equation (35) is the most efficient known exact method for finding the permanent of an arbitrary matrix. Its time complexity is $\mathcal{O}(n^2 2^n)$, which may be improved to $\mathcal{O}(n 2^n)$ by evaluating the trailing sum using a Gray-code order [38].

The running time is further reduced when \mathbf{M}^k possesses groups of repeated columns and/or rows. The permanent may then be computed from a smaller $\hat{n} \times \hat{m}$ -dimensional matrix $\hat{\mathbf{M}}^k$ that comprises only the unique columns and rows of \mathbf{M}^k . As shown in Appendix C, (35) may be recast as

$$\text{Per}(\mathbf{M}^k) = \sum_{\mathbf{d} \in \prod_{j=1}^{\hat{n}} \mathbb{N}_{n_j}} (-1)^{n + \|\mathbf{d}\|_1} \left[\prod_{j=1}^{\hat{n}} \binom{d_j}{n_j} \right] \prod_{q=1}^{\hat{m}} (\mathbf{d} \cdot \bar{\mathbf{M}}_q^k)^{m_q}, \quad (36)$$

where $\|\cdot\|_1$ is the L^1 norm, $\binom{d_j}{n_j}$ are binomial coefficients, $\prod_{j=1}^{\hat{n}} \mathbb{N}_{n_j}$ is a Cartesian product of sets of positive integers, and n_j and m_q are the number of identical

members in the j th repeated-column and q th repeated-row groups, respectively. The corresponding time complexity is given by

$$\mathcal{O} \left(\min \left(\hat{m} \prod_{j=1}^{\hat{n}} (n_j + 1), \hat{n} \prod_{q=1}^{\hat{m}} (m_q + 1) \right) \right), \quad (37)$$

where the minimization operation results from the permanent’s invariance under matrix transposition.

Repeated columns will be present in \mathbf{M}^k whenever identities aggregate into equivalence classes and/or there are repeated b_i^k (i.e., $|T| > |I|$). Similarly, \mathbf{M}^k will possess repeated rows in applications where there are two or more hidden tracks. For certain observability/equivalence-class conditions, the optimized Ryser formula of (36) may therefore render computable an otherwise intractable identity-deconflation step. In particular, (37) demonstrates that the computability of the modified Ryser method is quite favourable when either the number of observed targets or the number of target classes is moderate in size. In this regime, the computation time exhibits little dependence on the absolute number of identities. Finally, note that computation of (36) and (35) may be readily (and efficiently) parallelized in a manner that preserves the Gray-code evaluation sequence.

B. Numerical Considerations Concerning Exact Methods

Although the Ryser formula and its derivatives yield exact results for infinite-precision arithmetic, they entail computing the sum of terms of alternating sign, some of which may be of considerable magnitude. Therefore, implementation of these methods demands careful selection of numerical libraries and their parameters to ensure that the minimum requirements for arithmetic precision (e.g., the number of bits in the mantissas of floating point numbers) are satisfied. In many cases, data types based on native floating point implementations are inadequate, requiring the use of variable-precision libraries such as the GNU Multiple Precision (GMP) Arithmetic Library [42].

The smallest floating-point mantissa that safeguards numerical accuracy follows directly from the maximum possible ratio between intermediate summation terms and the permanent itself. Although the former is easily bounded from above by inspection of (35) and (36), non-trivial lower bounds for arbitrary positive matrices are generally not available. However, in view of the fact that a tight lower bound of e^{-n} exists for the permanent of doubly stochastic¹¹ matrices [43], it is beneficial analyze the factorization

$$\mathbf{M}_{\text{DS}}^k = \mathbf{D}_1^k \mathbf{M}^k \mathbf{D}_2^k, \quad (38)$$

where \mathbf{M}_{DS}^k is doubly stochastic, and \mathbf{D}_1 and \mathbf{D}_2 are invertible diagonal matrices possessing only non-negative

¹¹A doubly stochastic matrix possesses unit row and column sums.

entries. When each element of \mathbf{M}^k is greater¹² than some $\alpha > 0$, the existence of this decomposition is guaranteed by the Sinkhorn theorem¹³ [46]. Using elementary properties of the matrix permanent, (38) is easily shown to yield

$$\text{Per}(\mathbf{M}_{\text{DS}}^k) = \prod_{j=1}^n (\mathbf{D}_1^k)_{j,j} \prod_{j=1}^n (\mathbf{D}_2^k)_{j,j} \text{Per}(\mathbf{M}^k), \quad (39)$$

allowing calculation of $\text{Per}\mathbf{M}^k$ to be replaced with that of $\text{Per}\mathbf{M}_{\text{DS}}^k$. As the row sums of a doubly stochastic matrix are necessarily unity, an upper bound for the intermediate terms in both (35) and (36) may be taken as 2^n . The resulting ratio is thus $(2e)^n$, and the required size of the mantissa is given by

$$N_{\text{M}} = n(1 + \log_2 e) + N_+, \quad (40)$$

where N_+ denotes the number of additional bits determined by the number of significant digits required in the final result and the anticipated accumulation of round-off error.

The variable cost of floating-point operations will affect the computational requirements of the (modified) Ryser algorithm. Provided that the binomial coefficients and their inverses are precomputed, the sum of (36) may be calculated in a generalized Gray-code sequence using only additions, subtractions, and multiplications, the last of which dominates the asymptotic complexity (the same holds unmodified Ryser algorithm). Thus, (37) may be revised by scaling it with the prefactor $N_{\text{M}} \log N_{\text{M}} 2^{\mathcal{O}(\log^* N_{\text{M}})}$, which is the asymptotic complexity of integer multiplication using the Fürer algorithm¹⁴ [47]. As it is unlikely that the original data or the rescaling step will outstrip the limitations of native data types, enhanced precision should only be necessary when computing the permanent itself.

Computing the factorization of (38) may itself add to the total computational complexity of the (modified) Ryser method. For a residual L^∞ error of ϵ (over the row and column sums in \mathbf{M}_{DS}^k), the diagonal matrices may be computed approximately in $\mathcal{O}(n^4 \log(n/\epsilon) \log(1/\alpha))$ time using the RAS algorithm [48] or, alternatively, by iterative Sinkhorn scaling [46] at the cost of a slightly less favourable runtime. When $\epsilon < \alpha$, the permanent will deviate from that of its fully doubly-stochastic counterpart by at most a factor of $|(\alpha - \epsilon)/\alpha|^n$. To ensure that the commensurate change of accuracy requirements does not exceed one bit, the relation $0.5 < |(\alpha \pm \epsilon)/\alpha|^n < 2$ must be satisfied. By solving for ϵ at the extreme

¹²In accordance with Cromwell's rule [44], never allowing the Bayesian update step to produce a zero element in \mathbf{M}^k is generally good practice, and consequently, elements of \mathbf{M}^k are assumed to be positive.

¹³Actually, the strict positivity requirement can be relaxed for matrices that satisfy certain indecomposability conditions [45] [46].

¹⁴This is the fastest known integer multiplication algorithm and may be easily extended to floating point arithmetic.

points, expanding the solutions using a pair of Taylor series with respect to $1/n$, and bounding from above the sums with those of the associated infinite geometric series, it may be found that $\epsilon < (\alpha \ln 2)/n$ must hold asymptotically. The complexity of the RAS algorithm therefore becomes, $\mathcal{O}(n^4 \log(n/\alpha) \log(1/\alpha))$, which may be improved by processing the matrix that comprises only the unique columns and rows of \mathbf{M}^k .

Finally, it is observed that \mathbf{M}_{DS}^k may itself be used as an approximation to the matrix of marginal track-to-identity assignment probabilities \mathbf{P}^k (§2-C). Like \mathbf{P}^k , \mathbf{M}_{DS}^k possess unit row and column sums and may thus be regarded as a collection of quasi probabilities (though at the cost of an error that is difficult to characterize). This approach was used the basis of approximating the multitarget identification problem in [20]–[22].

C. Methods of Approximation

The challenge of the matrix permanent calculation rests with its membership in the #P-hard complexity class [49], whose problems are at least as computationally demanding as those of NP-hard. While the development of exact, polynomial-time algorithms thus seems unlikely, random-approximation methods for non-negative matrices appear considerably more promising and have drawn significant research interest. In particular, a fully polynomial-time randomized approximation scheme (FPRAS) for 0-1 and non-negative matrices was developed in [34]. This work, which constructed a Markov chain Monte-Carlo algorithm for sampling perfect matchings from the associated bipartite graph, exhibits a complexity¹⁵ of $\mathcal{O}^*(n^{26})$, a result that was subsequently improved to $\mathcal{O}^*(n^{10})$ [35] and then $\mathcal{O}^*(n^7)$ [50], [51]. Several other methods have been developed for matrices with additional properties. In particular, [52] uses the self-reducibility of the permanent to derive an algorithm whose complexity is $\mathcal{O}(n^4 \log^4 n)$ for matrices satisfying certain density requirements.

Loopy belief propagation has served as another basis for approximating the permanent of non-negative matrices [53]–[57]. This approach appears to exhibit encouraging accuracy and computational efficiency in real-world applications, but is presently understood in terms of heuristics that lack the theoretical performance guarantees enjoyed by the Monte-Carlo methods. Nonetheless, belief propagation remains a promising area of research, and significant ongoing attention has been directed towards establishing the theoretical underpinnings necessary for conducting rigorous analysis of accuracy and running time.

Finally, it should be noted that latent structure in the coefficients of \mathbf{M}_{DS}^k (such as sparsity) may afford further savings in computation time. The manner in which the aforementioned algorithms may be optimized under this condition is expected to depend significantly on the data

¹⁵ $\mathcal{O}^*(\cdot)$ denotes a complexity wherein logarithmic factors have been suppressed.

itself (and therefore the specific application). For this reason, investigations using empirical data from various problem domains may be valuable avenues of further research.

4. EXAMPLE PROBLEMS

The mathematical framework developed in the previous sections is illustrated by way of three examples (each a single run) that track and identify (or classify) a group of well-separated simulated identities moving independently in two dimensions.

A. Simulation Setup

1) Preliminaries:

Each identity's actual (time-indexed) state (\mathbf{x}_i^k) comprises position and velocity ($\text{pr}_d \mathbf{x}_i^k \in \mathbb{S}_{i,d} = \mathbb{R}^2 \times \mathbb{R}^2$) and a single, unchanging attribute ($\text{pr}_s \mathbf{x}_i^k \in \mathbb{S}_{i,s} = \mathbb{R}$) as dynamic and static components, respectively. The static attribute may be regarded as a feature that is observed in some of the example problems. The single-identity state space is therefore $\mathcal{S} = \mathbb{S}_{i,d} \times \mathbb{S}_{i,s} = \mathbb{R}^4 \times \mathbb{R}$, which is equipped with the product measure $\lambda \times \delta_{A_i}$ defined by the ordinary Lebesgue measure λ and the Dirac measure δ_{A_i} , where $A_i \in \mathbb{R}$ is the value of the identity's static attribute.

To simplify the remaining analysis, the detection and false alarm probabilities are set to 1 and 0, respectively, and a common number of tracks and identities is used. Thus, $|T_O \cup T_H| = |I|$, and each observed track will correspond to a single identity (and vice versa). The actual mapping between identities and observed targets is described by the permutation matrix

$$\mathbf{U}_{t,i} = \begin{cases} 1 & \text{If identity } i \text{ is actually at track } t \\ 0 & \text{Otherwise} \end{cases}, \quad (41)$$

which, along with the collection of time-indexed identity states \mathbf{x}_i^k , forms the reference 'ground truth' over the course of the simulations. Note that \mathbf{U} remains fixed over the course of each simulation.

2) Simulation of Observations and Target Motion:

Observations of the system, during which the positions and features (but not velocity) of all targets are simultaneously measured, are made at intervals of one second (where the relabeling given at the end of §2-B is used). Targets are assumed to be well separated, and each observed track $t \in T_O$ is unambiguously associated with a time series of measurements \mathbf{z}_t^k (indexed by $k = \{1, 2, \dots\}$) that were chosen to carry noisy position and feature information as

$$\mathbf{z}_t^k = \mathbf{H} \mathbf{x}_i^k + \begin{bmatrix} \mathfrak{M}_p \\ \mathfrak{M}_f \end{bmatrix}, \quad (42)$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 \end{bmatrix}, \quad (43)$$

i is the actual identity at track t satisfying the 'ground-truth' mapping $\mathbf{U}_{t,i} = \mathbf{I}$, \mathbf{I} is the 2×2 identity matrix, and the random samples \mathfrak{M}_p and \mathfrak{M}_f are drawn from the normally distributed bivariate position- and univariate feature-measurement noises M_p and M_f , respectively.

For a given identity i , transitions between an successive states (i.e., \mathbf{x}_i^k to \mathbf{x}_i^{k+1}) were based on the white-noise acceleration model [58]. For a unit time step, consecutive state vectors are related by

$$\mathbf{x}_i^{k+1} = \mathbf{F} \mathbf{x}_i^k + \begin{bmatrix} \mathfrak{P}_i/2 \\ \mathfrak{P}_i \\ 0 \end{bmatrix}, \quad (44)$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 \end{bmatrix}, \quad (45)$$

and \mathfrak{P}_i is a random sample of the bivariate normally distributed process noise P_i that is characteristic to the i th-identity (P_i is also stationary with respect to k). In general, the process noise (i.e., kinematics) is made to vary among identities such that $P_i \not\sim P_{i'}$ for $i \neq i'$, with the exception that $P_i \sim P_{i'}$ when i and i' are members of a common class of indistinguishable identities.

3) Implementation of Single-Target Filters:

Filtering at each observed track $t \in T_O$ is carried out in accordance with §2-B using a bank of $|I|$ filters that are perfectly matched to the actual measurement and transition processes given in (42) and (44). The Markov transition and measurement steps for the i th identity in a given track's filter bank are therefore modeled as

$$\mathbf{X}_i^{k+1} = \mathbf{F} \mathbf{X}_i^k + \begin{bmatrix} P_i/2 \\ P_i \\ \mathbf{0} \end{bmatrix} \quad (46)$$

and

$$\mathbf{Z}_i^k = \mathbf{H} \mathbf{X}_i^k + \begin{bmatrix} M_p \\ M_f \end{bmatrix}, \quad (47)$$

respectively.

Owing to the fact that P_i , M_p , and M_f are normal, the normality of any prior X_i^1 will be preserved in subsequent X_i^k , allowing the tracking to be implemented using linear Kalman filters. The density function defined in (20), which corresponds to the i th identity at the t th track, is thus given by

$$p_{t,i}^k(\mathbf{x}_i) = \frac{1}{\sqrt{(2\pi)^5 \text{Det}(\Sigma_{t,i}^k)}} e^{-\frac{1}{2}(\mathbf{x}_i - \mu_{t,i}^k)^T (\Sigma_{t,i}^k)^{-1} (\mathbf{x}_i - \mu_{t,i}^k)}, \quad (48)$$

where $\mu_{t,i}^k$ and $\Sigma_{t,i}^k$ are the density function's mean vector and covariance matrix, respectively. A collection of $p_{t,i}^k(\mathbf{x}_i)$ spanning every $i \in I$ is maintained by a separate filter bank for each observed track $t \in T_O$ (Fig. 2). The

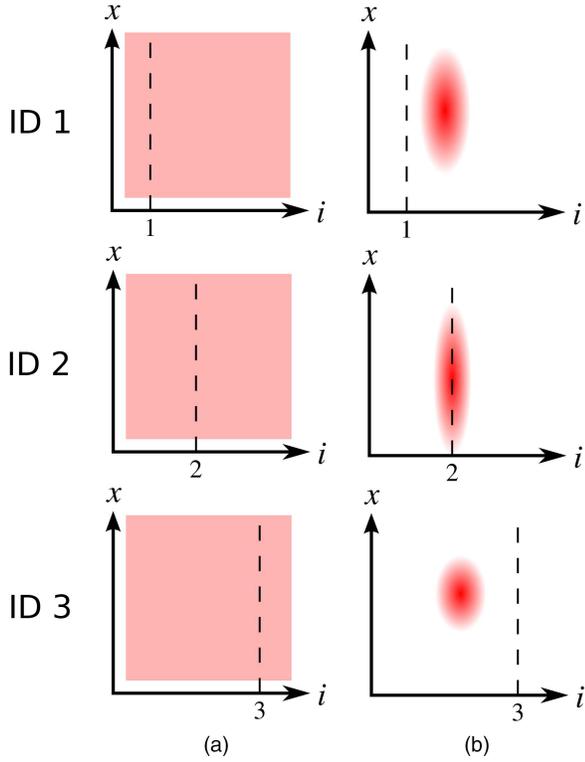


Fig. 2. Illustration of a filter bank's density functions ($p_{i,i}^k(\mathbf{x}_i)$), shown over three identities ($i = 1, 2, 3$) at some observed track $t \in T_O$ and timesteps (a) $k = 0$ (which is the prior and taken to be uniform) and (b) some $k > 0$. Note that the actual functions span additional kinematic dimensions not shown. The dashed lines represent the Dirac measures of (3) that are unique to each identity space and 'select' the probability associated with a particular identity ($i = 1, 2, 3$). Therefore, ID 2 represents the most probable identity in the filter bank of (b). As in JTC, there exist variations between the filter bank's individual densities that result from employing different Markov and measurement models with distinct identities (classes).

identity-specific, track-level transition and likelihood functions (defined in (22) and (26)) are realized as

$$f_{i, \mathcal{X}_i^{k+1} | \mathcal{X}_i^k}(\mathbf{x}_i | \mathbf{x}_i') = \frac{1}{\sqrt{(2\pi)^5 \text{Det}(\Sigma_{P_i}^k)}} e^{-\frac{1}{2}(\mathbf{x}_i - \mathbf{F}\mathbf{x}_i')^T (\Sigma_{P_i}^k)^{-1} (\mathbf{x}_i - \mathbf{F}\mathbf{x}_i')} \quad (49)$$

and

$$g_{i, \mathcal{Z}_i^k}(\mathbf{x}_i) = p_{i,t}^k + \frac{P_{d,i}^k}{\sqrt{(2\pi)^3 \text{Det}(\Sigma_{M_i}^k)}} e^{-\frac{1}{2}(\mathcal{Z}_i^k - \mathbf{H}\mathbf{x}_i)^T (\Sigma_{M_i}^k)^{-1} (\mathcal{Z}_i^k - \mathbf{H}\mathbf{x}_i)}, \quad (50)$$

with covariance matrices $\Sigma_{P_i}^k$ and $\Sigma_{M_i}^k$, respectively (recall that $p_{i,t}^k$ is zero). As each term in the first equation of (29) is Gaussian, the $K_{i,i}^k$ may be found analytically as

$$K_{i,i}^k = E \cdot \frac{1}{2} (\mu'^T \Sigma'^{-1} \mu' - \mu_{i,i}^{k,T} ((\mathbf{F}\Sigma_{P_i}^k \mathbf{F}^T)^{-1} + \Sigma_{P_i}^{-1}) \mu_{i,i}^k - \mathcal{Z}_i^{k,T} \Sigma_{M_i}^{-1} \mathcal{Z}_i^k) \quad (51)$$

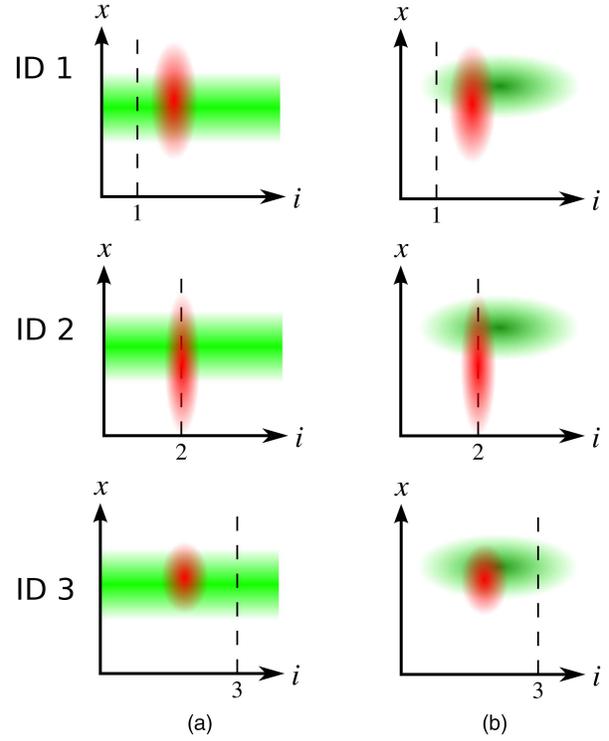


Fig. 3. Example relationship between the density functions ($p_{i,i}^k(\mathbf{x}_i)$, shown in red) of the filter bank at observed track $t \in T_O$ and a given Gaussian measurement likelihood function ($g_{i, \mathcal{Z}_i^k}(\mathbf{x}_i)$, shown in green), where the latter contains (a) only kinematic information and (b) both kinematic and identity information. In this example, those measurement functions also carrying identity information correspond to a noisy feature observation centred near ID 2. The update steps of $p_{i,i}^k(\mathbf{x}_i)$ in (27) and (28) are equivalent to marching the $p_{i,i}^k(\mathbf{x}_i)$ forward in time (in accordance with (49)) and, if the observation was associated to the filter bank's track, multiplying the results with the $g_{i, \mathcal{Z}_i^{k+1}}(\mathbf{x}_i)$. The updated densities are renormalized alongside the computation of the b_i^{k+1} and $a_{i,i}^{k+1}$ coefficients. Once again, the dashed lines represent the identity-specific Dirac measures, and the actual functions span additional kinematic dimensions not shown.

where

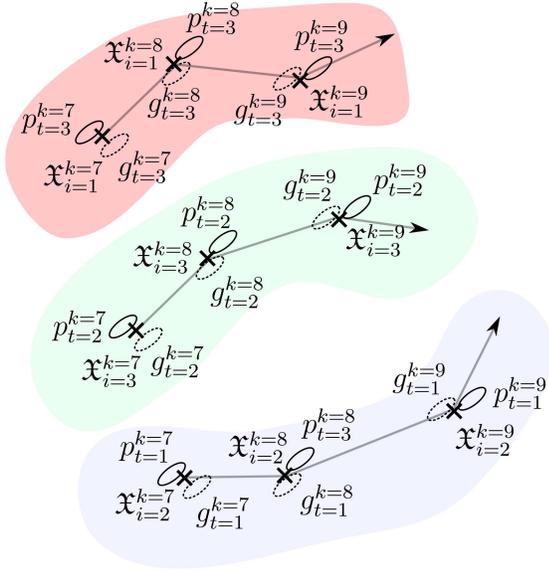
$$E = \sqrt{\frac{\text{Det}(\Sigma')}{(2\pi)^3 \text{Det}(\Sigma_{M_i}^k) \text{Det}((\Sigma_{t,i}^k)^{-1} + \mathbf{F}^T \Sigma_{P_i}^{-1} \mathbf{F})}} \quad (52)$$

$$\Sigma' = ((\Sigma_{t,i}^k)^{-1} + \mathbf{F}^T \Sigma_{P_i}^{-1} \mathbf{F} + \mathbf{H}^T \Sigma_{M_i}^{-1} \mathbf{H})^{-1}$$

$$\mu' = \Sigma' [((\Sigma_{t,i}^k)^{-1} \mathbf{F}^{-1} + \mathbf{F}^T \Sigma_{P_i}^{-1}) \mu_{i,i}^k + \mathbf{H}^T \Sigma_{M_i}^{-1} \mathcal{Z}_i^k].$$

Bayesian updates (which involve computing the $k + 1$ counterparts to the $p_{i,i}^k(\mathbf{x}_i)$, b_i^k , and $a_{i,i}^k$) are performed in accordance with (27) and (28) and are shown graphically in Fig. 3. At a given timestep k , the relationships between \mathfrak{X}_i^k (actual state of identity i), $p_{i,i}^k(\mathbf{x}_i)$ (i th identity density at track t), $g_{i, \mathcal{Z}_i^k}(\mathbf{x}_i)$ (i th identity measurement likelihood function at track t), and \mathbf{U} ('ground-truth' matrix that encodes the actual track-to-identity configuration) are illustrated in Fig. 4.

As a separate filter bank was maintained at each observed track, and a total of $|T_O| \cdot |I|$ individual filters



$$\begin{array}{c}
 I = \{1, 2, 3\} \\
 \xrightarrow{\hspace{1.5cm}} \\
 \mathbf{U} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \downarrow T_O = \{1, 2, 3\}
 \end{array}$$

Fig. 4. Illustration of the relationship between the actual identity positions (crosses), projected position densities (solid ellipses), and measurement densities (dashed ellipses), which are labelled as \mathcal{X}_i^k , p_t^k , and g_t^k , respectively, for the (arbitrarily chosen) timesteps $k = 7, 8, 9$. The latter are shorthand for the sets of functions $p_{t,i}^k(\mathbf{x}_i)$ (equation 48) and $g_{t,i}^k(\mathbf{x}_i)$ (equation 50), respectively. Similarly, each of their associated ellipses represents a set of ellipses indexed over I (recall that a bank of $|I|$ filters is run at every track, where each filter is programmed with the kinematic/static feature information of single $i \in I$). Remaining unknown to the estimation algorithms, the matrix \mathbf{U} (equation 41)—shown here as the realization induced by an arbitrary order on I and T_O —contains the ‘ground-truth’ associations between identities and tracks that define a given simulation. Note that the velocity and static attribute components of p_t^k and \mathcal{X}_i^k are not shown.

were employed. In every example, targets were assigned a zero initial velocity, and a uniform prior was used in the remaining dimensions by setting $p_{t,i}^1(\mathbf{x}) = \mathbf{1}_r(\mathbf{x})$ and $\mathbf{M}_{t,i}^1 = 1$ for all t and i . The $\mathbf{P}_{t,i}^k$ were calculated with a parallel implementation of the speed-improved Ryser formula of §3-A using the GMP Arithmetic Library. In accordance with §3-B, the minimum significant size was found to be $n(1 + \log_2 e) \approx 244$, to which another 56 bits was added to furnish a working precision of approximately 16 significant digits.¹⁶ Prior to calculating the permanent, small values of $a_{t,i}^k$ were rounded up to 10^{-100} , and the \mathbf{M}^k were converted to doubly stochastic-

¹⁶Repeating the simulations with 10,000-bit mantissas yielded no significant change in the final results.

tic form using Sinkhorn scaling. The maximum-weight matchings were found with an $\mathcal{O}(n^3)$ implementation of the Hungarian method [59], and all processing was performed on an HP Z820 workstation.

When each identity is uniquely resolvable, the matrix of track-to-identity assignment probabilities \mathbf{P}^k will asymptotically converge to \mathbf{U} over the subset of observed tracks (i.e., $\mathbf{P}_{t,i}^k \rightarrow \mathbf{U}_{t,i}$ with increasing k for $t \in T_O$), given adequate measurements and an initial prior that satisfies Cromwell’s rule. However, when groups of targets are indistinguishable (e.g. common attributes and/or motion models), this is no longer holds. Assuming that identities of a given equivalence class are assigned equal initial values in each row of \mathbf{M}^1 , the asymptotic matrix of track-to-identity assignment probabilities becomes

$$\mathbf{V}_{t,i} = \begin{cases} 1/|\mathcal{I}(i)| & \text{If any } i' \in \mathcal{I}(i) \text{ is at } t \\ 0 & \text{Otherwise} \end{cases}, \quad (53)$$

where $\mathcal{I}(i) \subseteq I$ is the set of identities¹⁷ that are indistinguishable from i . In cases where $|\mathcal{I}(i)| = 1$, adequate measurements will perfectly locate i , with the special case $\mathbf{V} = \mathbf{U}$ when this holds for all $i \in I$.

4) Comparisons to Approximate Methods:

The Bayesian algorithm developed in the previous sections was evaluated by computing the true posterior track-to-identity assignment probabilities $\mathbf{P}_{t,i}^k$ and comparing the results with the asymptotically-optimal assignments encoded in \mathbf{V}^k . Further comparisons were carried out against the track-to-identity assignments computed by approximation methods discussed in §1. The corresponding approximate assignment matrices (containing quasi posterior probabilities) are:

1) The local (track-level) weights given by

$$\tilde{\mathbf{P}}_{\text{TL}}^k = \mathbf{D}^k \mathbf{M}^k, \quad (54)$$

where the diagonal matrix \mathbf{D}^k normalizes the rows of \mathbf{M}^k (i.e., $(\mathbf{D}_{t,t}^k)^{-1} = \sum_{i \in I} \mathbf{M}_{t,i}^k$). The elements in the t th row of \mathbf{M}^k are identity/class coefficients maintained by the filter bank running at the t th track, and the scaling effected by $\mathbf{D}_{t,t}^k$ is the post-Bayesian-update normalization that arises by treating the t th track as a single-target JTC problem. The elements of $\tilde{\mathbf{P}}_{\text{TL}}^k$ are thus the track-to-identity assignment probabilities computed by separate instances of single-target JTC (e.g. [9]–[11]) applied to each track.

2) The diagonally-scaled track-level weights given by

$$\tilde{\mathbf{P}}_{\text{DS}}^k = \mathbf{D}_1^k \mathbf{M}^k \mathbf{D}_2^k, \quad (55)$$

where, as discussed in §3-B, \mathbf{D}_1^k and \mathbf{D}_2^k are positive diagonal matrices that yield a doubly-stochastic $\tilde{\mathbf{P}}_{\text{DS}}^k$. This approximation was used in [20]–[22].

¹⁷Note that $i \in \mathcal{I}(i)$ and $\mathcal{I}(i') = \mathcal{I}(i)$ for all $i' \in \mathcal{I}(i)$.

TABLE I
Parameters of the example problem in §4-B.1

Description	Parameter	Value
Covariance (Process Noise)	Σ_i^P	$\begin{bmatrix} 0.25\mathbf{I} & 0.5\mathbf{I} & \mathbf{0} \\ [i/2] & 0.5\mathbf{I} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 \end{bmatrix}$
Mean (Process Noise)	μ_i^P	$\mathbf{0}$
Covariance (Meas. Noise)	$\Sigma_{t,i}^M$	$\begin{bmatrix} 4\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \infty\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 4 + \infty((t-1) \bmod 2) \end{bmatrix}$
Mean (Measurement Noise)	$\mu_{t,i}^M$	$\mathbf{0}$
Static Attribute	A_i	i

Note that i , t , \mathbf{I} , and $\lfloor \cdot \rfloor$ are the identity number (1–100), track number (1–6), 2×2 identity matrix, and floor function, respectively. By definition, the process noise covariance is shared by pairs of successive identities, while the measurement noise covariance matrix renders static attributes unobservable in every second track. This may be seen by noting that the static attribute variance given by $4 + \infty((t-1) \bmod 2)$ evaluates to 4 and ∞ for odd and even t , respectively, where the latter case is completely non-informative.

3) The maximum-weight matching $\tilde{\mathbf{P}}_{\text{MW}}^k$ defined as

$$(\tilde{\mathbf{P}}_{\text{MW}}^k)_{t,i} = \begin{cases} 1, & i = \sigma_{\text{MW}}^k(t) \\ 0, & i \neq \sigma_{\text{MW}}^k(t) \end{cases}, \quad (56)$$

where

$$\sigma_{\text{MW}}^k = \arg \max_{\sigma \in S_{|T|}} \prod_{t \in T} \mathbf{M}_{t,\sigma(t)}^k \quad (57)$$

and $S_{|T|}$ is the symmetric group of degree $|T|$. The matrix $\tilde{\mathbf{P}}_{\text{MW}}^k$ serves as a (hard) maximum likelihood track-to-identity estimate over the set of all possible assignments and was used in [18], [19].

Discrepancies between the (quasi) probability distributions were quantified as the maximum Kullback-Liebler (KL) divergence over all observed tracks

$$\max_{t \in T} D_{\text{KL}}(\alpha_t \| \beta_t) = \max_{t \in T} \sum_{i \in I} \ln \left(\frac{\alpha_{t,i}}{\beta_{t,i}} \right) \alpha_{t,i}, \quad t \in T_0 \quad (58)$$

for the ordered pairs of matrix rows $(\alpha_t, \beta_t) = (\mathbf{V}_t, \mathbf{P}_t^k)$, $(\mathbf{V}_t, \mathbf{a}_{\text{TL}}^k)$, $(\mathbf{V}_t, \mathbf{P}_{\text{DS}}^k)$, and $(\mathbf{V}_t, \tilde{\mathbf{P}}_{\text{MW}}^k)$. Note that the summand in (58) yields the information gain (in nats) realized by substituting α_t for β_t as the set of identity assignment probabilities at track t .

B. Simulation Results

1) 100 Identities, 6 Observed Tracks, Incomplete Measurements:

In this example, $\mathcal{S} = \mathcal{S}_{\cdot,d} \times \mathcal{S}_{\cdot,s} = \mathbb{R}^4 \times \mathbb{R}$, $|T| = 100$, $|T_0| = 6$, and $\mathbf{U} = \mathbf{I}$ (the identity matrix). The kinematics, static attributes, and measurement parameters of the identities are given in Table I, where tracks indexed 0–5

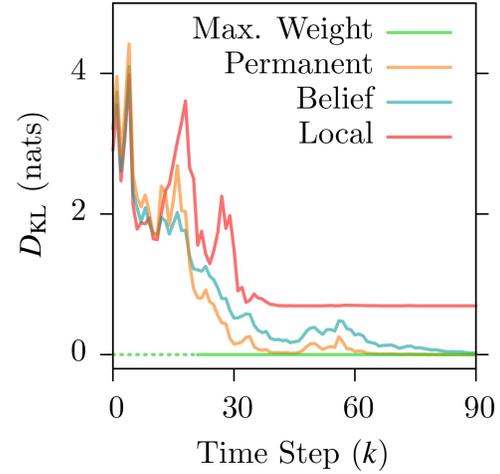


Fig. 5. Maximum Kullback-Liebler divergence (D_{KL}) over T_0 (the set of observed tracks) as a function of time step for the problem of Table I. The maximum weight divergence is undefined for $k < 22$ (as denoted by a broken line). Note that only the local (track-level) assignment fails to converge, demonstrating that identity resolution can require non-local information.

were made observable (by virtue of the fact that $\mathbf{U} = \mathbf{I}$, only the identities indexed 0–5 were actually observed). By definition, identities are endowed with unique static attributes but possess pairwise-common kinematic parameters (distinct kinematic characteristics are assigned to groups of two consecutively numbered identities). Furthermore, as the static attributes are only made observable for every second track, half of the single-target tracks should display persistent local identity ambiguities that fail to resolve with additional measurements. However, as the static attribute of one member in each pair of tracks is observable, *global* identity deconfliction will asymptotically resolve each track’s identity exactly.

The results of the simulation (which required 150 ms per timestep) are given in Fig. 5, which shows that the maximum KL divergences generally decrease with time step. Each of the global methods (permanent, belief matrix, and maximum-weight matching) converges to the correct identity-track permutation, as evidenced by respective maximum divergences that tend to zero. However, the maximum KL divergence computed using the local track probabilities converges to about 0.7 nats, reflecting the fact that the worst-performing local identifications—which occur for those tracks lacking observations of the static attribute—assign probabilities of ~ 0.5 to two identities that share the same kinematic properties. As expected, the statistically optimal matrix permanent algorithm outperforms the other soft assignment methods (which produce quasi probabilities). Finally, note that while the maximum KL divergence corresponding to the (hard) maximum-weight matching abruptly transitions from undefined to zero at $k = 22$, the soft assignments display more gradual convergence, behaviour that is broadly consistent with the differences between the respective classes of algorithms.

TABLE II
Parameters of the example problem in §4-B.2

Description	Parameter	Value
Covariance (Process Noise)	Σ_i^P	$\begin{bmatrix} 0.25\mathbf{I} & 0.5\mathbf{I} & \mathbf{0} \\ [i/2] & 0.5\mathbf{I} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 \end{bmatrix}$
Mean (Process Noise)	μ_i^P	$\mathbf{0}$
Covariance (Measurement Noise)	$\Sigma_{r,i}^M$	$\begin{bmatrix} 4\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \infty\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 4 \end{bmatrix}$
Mean (Measurement Noise)	$\mu_{r,i}^M$	$\mathbf{0}$
Static Attribute	A_i	i

With the exception of the measurement noise covariance, which is defined in a manner that makes static attribute information visible in every track, these parameters are identical to those of the first example (Table I).

2) 100 Identities, 6 Observed Tracks, Complete Measurements:

This problem is a variation on the previous example, differing only by the measurement noise covariance (Table II), which now extends static attribute visibility to all observed tracks. In this case, both the local and global track-to-identity assignments should asymptotically converge to the actual track-to-identity configurations. Nonetheless, the global assignments are expected to exhibit improved pre-asymptotic characteristics, which may be germane to applications that require interim track-to-identity estimates (or simply do not run to convergence).

The KL divergences of this simulation (which also required 150 ms per timestep) are shown in Fig. 6. As expected, each of the maximum KL divergences tends to zero with increasing time step. However, between $k = 0$ and $k = 30$, there is significant discrepancy between the local and global identity assignments, supporting the assertion that global identity deconfliction improves target identification, even when identities are locally resolvable. As in the previous example, the permanent-based algorithm exhibits the fastest soft convergence, and the maximum-weight matching finds the correct assignment (in this case for $k \geq 9$). Finally, note the significant improvement in convergence rates as compared to that of §4-B.1, which may be ascribed to the information gained from doubling attribute measurements.

3) 100 Identities, 3 Identity Equivalence Classes:

In this example, $\mathcal{S} = \mathbb{S}_{\cdot,d} = \mathbb{R}^4$, $|I| = 100$, $|T_0| = 100$, and $\mathbf{U} = \mathbf{I}$. No static properties are visible, and identity dynamics are divided into three equivalence classes given in Table III (each identity’s Markov process is described by one of three motion models). The results of this simulation (which required 270 ms per timestep) are given in Fig. 7, which shows that global deconfliction significantly improves the resolution of equivalence

TABLE III
Parameters of the example problem in §4-B.3

Description	Parameter	Value
Covariance (Process Noise)	Σ_i^P	$i \begin{bmatrix} 0.25\mathbf{I} & 0.5\mathbf{I} \\ 0.5\mathbf{I} & \mathbf{I} \end{bmatrix}$
Mean (Process Noise)	μ_i^P	$\mathbf{0}$
Covariance (Measurement Noise)	$\Sigma_{r,i}^M$	$\begin{bmatrix} 4\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \infty\mathbf{I} \end{bmatrix}$
Mean (Measurement Noise)	$\mu_{r,i}^M$	$\mathbf{0}$
No. identities in Equivalence Class 1	$n(i = 1)$	60
No. identities in Equivalence Class 2	$n(i = 2)$	30
No. identities in Equivalence Class 3	$n(i = 3)$	10

The subscript i indexes the equivalence class. Note the absence of static attribute information in this example.

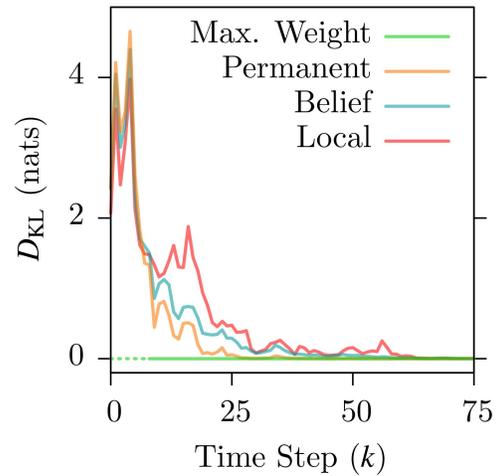


Fig. 6. Maximum Kullback-Liebler divergence (D_{KL}) over T_0 as a function of time step for the problem of Table II. Once again, the maximum weight divergence is undefined for $k < 9$ (as denoted by a broken line).

class after $k \sim 40$. Interestingly, the matrix-permanent algorithm performs only modestly better than the belief-matrix method, although both methods significantly outperform local assignment. Finally, in this example, convergence of the maximum-weight method appears substantially more uneven, finding the correct identity-track assignment at $k = 44$, then reverting to incorrect assignments at $k = 52$ and $k = 65$ before finally settling on correct assignment for $k \geq 66$.

5. CONCLUSION

This paper derived a rigorously Bayesian method for finding the optimal track-to-identity assignments for a group of targets that are well-separated or tracked using (J)PDA. Identification of targets is performed jointly across tracks to correctly account for the complex sta-

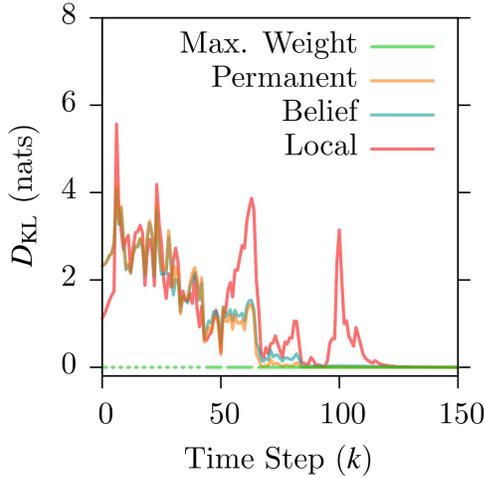


Fig. 7. Maximum Kullback-Liebler divergence (D_{KL}) over T_0 as a function of time step for the problem of Table III. The maximum weight divergence is undefined for $k < 44$ and again for $k = 52$ and $k = 65$ (as denoted by broken lines).

tistical dependencies between track-to-identity assignments. The number of tracks need not equal the number of identities, and arbitrary feature and kinematic measurements may be used, provided that their corresponding sensor models can be characterized statistically. The problem naturally decomposes into local single-target tracking and classification and global combinatorial identity deconfliction, where the former is based on a unified measure-theoretic framework that treats tracking and classification on equal footing, and the latter reduces to computing the permanent of a non-negative matrix. While the computational complexity of the matrix permanent poses challenging implementation issues, Markov chain Monte Carlo methods may be used to find approximations in polynomial time. Furthermore, the existence of groups of targets that are indistinguishable, unobservable, or both allows the Ryser formula to be modified in a manner that improves computation speed. Reducing the complexity of approximating non-negative matrix permanents is an area of significant contemporary research, and advances in this field will directly benefit the performance of the algorithm described in this work.

ACKNOWLEDGEMENTS

The author expresses gratitude to D. J. Peters, G. R. Mellema, A. W. Isenor, and M. Farrell (DRDC, Atlantic Research Centre) for constructive comments and discussions during the preparation of this manuscript.

APPENDIX A MARKOV TRANSITION AND BAYESIAN UPDATE

Under conditions 1–5 of §2-B, the Markov transition step of the general filtering problem (BF.2) may be

expanded as

$$\begin{aligned}
 & p_{X^k|Z^{k-1:1}}(\mathbf{x} | \mathfrak{Z}^{k-1:1}) \\
 &= \int_{\mathcal{S}} f_{X^k|X^{k-1}}(\mathbf{x} | \mathbf{x}') p_{X^{k-1}|Z^{k-1:1}}(\mathbf{x}' | \mathfrak{Z}^{k-1:1}) d\mu(\mathbf{x}') \\
 &= \int_{\mathcal{S}} \left[\prod_{i \in I} \sum_{t \in T} f_{t, X_i^k | X_i^{k-1}}(\mathbf{x}_i | \mathbf{x}'_i) \right] \left[C^{k-1} D^{k-1} \right. \\
 &\quad \cdot \mathbf{1}_{A^{k-1}}(\mathbf{x}) \prod_{i \in I} \sum_{t \in T} (b_t^{k-1})^{-1} a_{t,i}^{k-1} p_{t,i}^{k-1}(\mathbf{x}'_i) \left. \right] d\mu(\mathbf{x}'). \tag{59}
 \end{aligned}$$

Collecting product terms and noting the support of the $f_{t, X_i^k | X_i^{k-1}}(\mathbf{x}_i | \mathbf{x}'_i)$ and $p_{t,i}^{k-1}(\mathbf{x}'_i)$ —given in (22) and (20), respectively—yields the simplification

$$\begin{aligned}
 p_{X^k|Z^{k-1:1}}(\mathbf{x} | \mathfrak{Z}^{k-1:1}) &= C^{k-1} D^{k-1} \\
 &\cdot \mathbf{1}_{A^k}(\mathbf{x}) \prod_{i \in I} \sum_{t \in T} (b_t^{k-1})^{-1} a_{t,i}^{k-1} p_{t,i}^{k-0.5}(\mathbf{x}_i), \tag{60}
 \end{aligned}$$

where

$$p_{t,i}^{k-0.5}(\mathbf{x}_i) = \int_{r_t^{k-1}} f_{X_i^k | X_i^{k-1}}(\mathbf{x}_i | \mathbf{x}'_i) p_{t,i}^{k-1}(\mathbf{x}'_i) d\mu(\mathbf{x}'_i) \tag{61}$$

is the forward-in-time projection of $p_{t,i}^k(\mathbf{x}_i)$. The Bayesian update step may be similarly found by substituting (20) and (25) into the right-hand side of (BF.1) to produce

$$\begin{aligned}
 & p_{X^k|Z^{k:1}}(\mathbf{x} | \mathfrak{Z}^{k:1}) \\
 &\propto f_{Z^k|X^k}(\mathfrak{Z}^k | \mathbf{x}) p_{X^k|Z^{k-1:1}}(\mathbf{x} | \mathfrak{Z}^{k-1:1}) \\
 &= \left[p_f^k \mathbf{1}_{A^k}(\mathbf{x}) \prod_{i \in I} (\mathbf{1}_{r_i^k}(\mathbf{x}_i) (p_f^k)^{-1} g_{i,3^k}(\mathbf{x}_i) \right. \\
 &\quad \left. + \mathbf{1}_{C_i^k}(\mathbf{x}_i)) \right] \\
 &\quad \cdot \left[C^{k-1} D^{k-1} \mathbf{1}_{A^k}(\mathbf{x}) \prod_{i \in I} \sum_{t \in T} (b_t^{k-1})^{-1} \right. \\
 &\quad \left. \cdot a_{t,i}^{k-1} p_{t,i}^{k-0.5}(\mathbf{x}_i) \right]. \tag{62}
 \end{aligned}$$

Using (21), equation (62) may be rewritten as

$$\begin{aligned}
 & p_{X^k|Z^{1:k}}(\mathbf{x} | \mathfrak{Z}^{1:k}) = \\
 & C^{k-1} (D^{k-1} p_f^k) \mathbf{1}_{A^k}(\mathbf{x}) \\
 & \cdot \prod_{i \in I} \left[\left(\sum_{t \in T \setminus \{\mathbf{o}(k)\}} (b_t^{k-1})^{-1} a_{t,i}^{k-1} p_{t,i}^{k-1}(\mathbf{x}_i) \right) + \right. \\
 & \left. (b_{\mathbf{o}(k)}^{k-1} p_f^k)^{-1} \left(a_{\mathbf{o}(k),i}^{k-1} \int_{r_{\mathbf{o}(k)}} p_{\mathbf{o}(k),i}^{k-0.5}(\mathbf{x}'_i) g_{i,3^k}(\mathbf{x}_i) d\mu(\mathbf{x}'_i) \right) \right. \\
 & \left. \cdot \left(\frac{p_{\mathbf{o}(k),i}^{k-0.5}(\mathbf{x}_i) g_{i,3^k}(\mathbf{x}_i)}{\int_{r_{\mathbf{o}(k)}} p_{\mathbf{o}(k),i}^{k-0.5}(\mathbf{x}'_i) g_{i,3^k}(\mathbf{x}'_i) d\mu(\mathbf{x}'_i)} \right) \right]. \tag{63}
 \end{aligned}$$

The terms $p_{t,i}^k(\mathbf{x}_t)$, b_t^k , and $a_{t,i}^k$ are thus related to their counterparts at the previous time step by (27) and (28) for $t = \mathbf{o}(k)$ and $t \neq \mathbf{o}(k)$, respectively. In particular, note that only the coefficients associated with the measured track $t = \mathbf{o}(k)$ are updated.

APPENDIX B EVALUATION OF POSTERIOR ASSIGNMENT PROBABILITIES

A. Track-to-Identity Probabilities as a Ratio of Permanents

The posterior track-to-identity assignment probabilities $\mathbf{P}_{t,i}^k$ are found by integrating the i th-identity marginalization of state's joint density function over the region of \mathbb{S} associated with the t th track. This may be expanded with (20) as

$$\begin{aligned} \mathbf{P}_{t,i}^k &= \int_{r_t^k} p_{X_t^k}(\mathbf{x}'_t) d\mu(\mathbf{x}'_t) \\ &= \int_{r_t^k} \int_{\mathbb{S}_{I \setminus \{i\}}} p_{X_t^k}(\mathbf{x}') d\mu(\mathbf{x}') d\mu(\mathbf{x}'_{\mathbb{S}_{I \setminus \{i\}}}) \\ &= \int_{r_t^k} \int_{\mathbb{S}_{I \setminus \{i\}}} C^k D^k \mathbf{1}_{A^k}(\mathbf{x}') \\ &\quad \cdot \prod_{i' \in I} \sum_{i' \in T} (b_{i'}^k)^{-1} a_{i',i'}^k p_{i',i'}^k(\mathbf{x}'_{i'}) d\mu(\mathbf{x}'). \end{aligned} \quad (64)$$

Noting the decompositions of the indicator function given in (17) and (18), equation (64) becomes

$$\begin{aligned} \mathbf{P}_{t,i}^k &= C^k D^k \int_{r_t^k} \int_{\mathbb{S}_{I \setminus \{i\}}} \sum_{\sigma \in P(I,T)} \prod_{i'' \in I} \mathbf{1}_{\sigma(i'')}(\mathbf{x}'_{i''}) \\ &\quad \cdot \prod_{i' \in I} \sum_{i' \in T} (b_{i'}^k)^{-1} a_{i',i'}^k p_{i',i'}^k(\mathbf{x}'_{i'}) d\mu(\mathbf{x}') \\ &= C^k D^k \int_{r_t^k} \int_{\mathbb{S}_{I \setminus \{i\}}} \sum_{\sigma \in P(I,T)} \prod_{i' \in I} \sum_{i' \in T} (b_{i'}^k)^{-1} \\ &\quad \cdot a_{i',i'}^k \mathbf{1}_{\sigma(i')}(\mathbf{x}'_i) p_{i',i'}^k(\mathbf{x}'_i) d\mu(\mathbf{x}'). \end{aligned} \quad (65)$$

Using (21), this simplifies to

$$\begin{aligned} \mathbf{P}_{t,i}^k &= C^k D^k \int_{r_t^k} \int_{\mathbb{S}_{I \setminus \{i\}}} \sum_{\sigma \in P(I,T)} \prod_{i' \in I} (b_{\sigma(i')}^k)^{-1} \\ &\quad \cdot a_{\sigma(i'),i'}^k p_{\sigma(i'),i'}^k(\mathbf{x}'_i) d\mu(\mathbf{x}'). \end{aligned} \quad (66)$$

As the probability densities in (66) have unit-valued integrals, integration with respect to $\mathbf{x}_{\mathbb{S}_{I \setminus \{i\}}}$ yields

$$\begin{aligned} \mathbf{P}_{t,i}^k &= C^k D^k \int_{r_t^k} \sum_{\sigma \in P(I,T)} p_{\sigma(i),i}^k(\mathbf{x}'_i) \prod_{i' \in I} (b_{\sigma(i')}^k)^{-1} \\ &\quad \cdot a_{\sigma(i'),i'}^k d\mu(\mathbf{x}'_i). \end{aligned} \quad (67)$$

The final integral zeros all terms not indexed by $\sigma(i) = t$, giving

$$\mathbf{P}_{t,i}^k = C^k D^k \sum_{\substack{\sigma \in P(I,T) \\ \sigma(i)=t}} \prod_{i' \in I} (b_{\sigma(i')}^k)^{-1} a_{\sigma(i'),i'}^k. \quad (68)$$

The constant C^k may be found by solving

$$\begin{aligned} 1 &= \int_{\mathbb{S}} p_{X^k}(\mathbf{x}') d\mu(\mathbf{x}') \\ &= \int_{\mathbb{S}} C^k D^k \mathbf{1}_{A^k}(\mathbf{x}') \\ &\quad \cdot \prod_{i' \in I} \sum_{i' \in T} (b_{i'}^k)^{-1} a_{i',i'}^k p_{i',i'}^k(\mathbf{x}'_{i'}) d\mu(\mathbf{x}') \end{aligned} \quad (69)$$

for $(C^k)^{-1}$. With the exception of the final integral, this process mirrors the steps of (64)–(68), giving

$$C^k = D^k \sum_{\sigma \in P(I,T)} \prod_{i' \in I} (b_{\sigma(i')}^k)^{-1} a_{\sigma(i'),i'}^k. \quad (70)$$

Noting that T and I are endowed with total orders, $t \in T$ and $i \in I$ may be used to index the elements of a $|T| \times |I|$ -dimensional matrix defined as

$$\mathbf{A}_{t,i}^k = a_{t,i}^k. \quad (71)$$

Similarly, a $|T| \times |T|$ diagonal matrix may be defined as

$$\mathbf{B}_{t,j}^k = \text{diag}(b_t^k) = \begin{cases} b_t^k, & t = j \\ 0, & t \neq j \end{cases}, \quad (72)$$

where j has been used in lieu of i , given that $i \leq |I| \leq |T|$ (as described in §2-A). Equation (68) thus becomes

$$\mathbf{P}_{t,i}^k = C^k D^k \sum_{\substack{\sigma \in P(I,T) \\ \sigma(i)=t}} \prod_{i' \in I} ((\mathbf{B}^k)^{-1} \mathbf{A}^k)_{\sigma(i'),i'}. \quad (73)$$

A new $|T| \times |T|$ -dimensional augmented matrix may be constructed as

$$[(\mathbf{B}^k)^{-1} \mathbf{A}^k \mid \mathbf{J}], \quad (74)$$

where \mathbf{J} is an $|T| \times (|T| - |I|)$ -dimensional matrix of ones. Note that this matrix is empty when $|T| - |I| = 0$. Reformulated with respect to $[(\mathbf{B}^k)^{-1} \mathbf{A}^k \mid \mathbf{J}]$, equation (73) becomes

$$\mathbf{P}_{t,i}^k = \frac{C^k D^k}{(|T| - |I|)!} \sum_{\substack{\sigma \in S_{|T|} \\ \sigma(i)=t}} \prod_{j=1}^{|T|} [(\mathbf{B}^k)^{-1} \mathbf{A}^k \mid \mathbf{J}]_{\sigma(j),j}, \quad (75)$$

where $S_{|T|} = P(\mathbb{N}_{|T|}, \mathbb{N}_{|T|})$ is the symmetric group of degree $|T|$, and $(|T| - |I|)!$ is the number of terms in $S_{|T|}$ that are associated with a single term in (the smaller) $P(I, T)$. Using the second equation in (29) and noting that D^k may be brought under the sum product by scaling the rows of $[(\mathbf{B}^k)^{-1} \mathbf{A}^k \mid \mathbf{J}]$ by \mathbf{B}^k , equation (69) may be rewritten as

$$\begin{aligned} \mathbf{P}_{t,i}^k &= \frac{C^k}{(|T| - |I|)!} \sum_{\substack{\sigma \in S_{|T|} \\ \sigma(i)=t}} \prod_{j=1}^{|T|} [\mathbf{A}^k \mid \mathbf{B}^k \mathbf{J}]_{\sigma(j),j} \\ &= \frac{C^k}{(|T| - |I|)!} \sum_{\substack{\sigma \in S_{|T|} \\ \sigma(i)=t}} \prod_{j=1}^{|T|} \mathbf{M}_{\sigma(j),j}^k, \end{aligned} \quad (76)$$

where \mathbf{M}^k is defined as

$$\mathbf{M}^k = [\mathbf{A}^k \mid \mathbf{B}^k \mathbf{J}]. \quad (77)$$

Finally, (76) may be simplified to

$$\mathbf{P}_{t,i}^k = \frac{C^k}{(|T| - |I|)!} \mathbf{M}_{t,i}^k \sum_{\sigma \in \mathcal{S}_{|T|-1}} \prod_{j=1}^{|T|-1} \mathbf{M}_{\sigma(j),j}^k(t;i), \quad (78)$$

where $\mathbf{M}^k(t;i)$ is the matrix formed from \mathbf{M}^k by deleting the t th row and i th column. Equation (70) may be similarly expressed as

$$C^k = \frac{1}{(|T| - |I|)!} \sum_{\sigma \in \mathcal{S}_{|T|}} \prod_{j=1}^{|T|} \mathbf{M}_{\sigma(j),j}^k. \quad (79)$$

Each of the combinatorial sums in (78) and (79) conforms to a matrix permanent defined in (32), and marginal probability in (78) thus becomes the ratio

$$\mathbf{P}_{t,i}^k = \mathbf{M}_{t,i}^k \frac{\text{Per}(\mathbf{M}^k(t;i))}{\text{Per}(\mathbf{M}^k)}. \quad (80)$$

While the resulting matrix of marginal probabilities \mathbf{P}^k is at least left stochastic, it becomes doubly stochastic when every track is certain to exist, a condition that results when each identity can be found at some track and vice versa. In this case, $|T| = |I|$, \mathbf{J} is empty (the b_t coefficients are consequently irrelevant), and each track must therefore exist and correspond to some identity—regardless of the system's false alarm probabilities. When this is undesired, a set of hidden tracks may be defined in accordance with (1) of §2-A. Finally, note that groups of identities that are indistinguishable with respect to the suite of sensors form equivalence classes, the presence of which is manifested by repeated columns in \mathbf{M}^k .

B. Limiting Process for Zero False-Alarm Probabilities

The zero false-alarm probabilities that were temporarily reassigned non-zero values (Condition 5 in §2-B) may now be taken to zero in the limit. In what follows, the $p_f^{k' \leq k}$ in \mathbf{M}^k of (78) are sequentially brought to zero for each $p_f^{k'}$ for which a limit is required. This process commences by recursively expanding (BF.1) and (BF.2) over all $k'' \leq k$ with the measurement function defined in (24). For any $p_f^{k'}$ undergoing the limiting step, this yields

$$\begin{aligned} \lim_{p_f^{k'} \rightarrow 0^+} p_{X^k|Z^{1:k}}(\mathbf{x}^{k'} \mid \mathfrak{Z}^{1:k}) &= \\ \lim_{p_f^{k'} \rightarrow 0^+} \cdots \int_{\mathbb{S}} \mathbf{1}_{A^k}(\mathbf{x}^{k'}) \prod_{i \in I} & \\ (\mathbf{1}_{r_i^{k'}}(\mathbf{x}_i^{k'}) (p_f^{k'})^{-1} g_{i,3^{k'}}(\mathbf{x}_i^{k'}) + \mathbf{1}_{C_{r_i^{k'}}}(\mathbf{x}_i^{k'})) & \\ \cdots d\mu(\mathbf{x}^{1:k-1}), & \end{aligned} \quad (81)$$

where $1 \leq k' \leq k$ and the ellipses contain the measurement functions for $k'' \neq k'$ and the Markov transition densities and their associated integrals for all $k'' \leq k$.

Employing the equivalence between (23) and (24) results in the reformulation

$$\begin{aligned} \lim_{p_f^{k'} \rightarrow 0^+} p_{X^k|Z^{1:k}}(\mathbf{x} \mid \mathfrak{Z}^{1:k}) &= \lim_{p_f^{k'} \rightarrow 0^+} \left(\cdots \int_{\mathbb{S}} \mathbf{1}_{A^{k'}}(\mathbf{x}^{k'}) \right. \\ &\cdot \left. \left(\sum_{i \in I} p_{d,i}^{k'} f_{Z^{k'}|X_S^{k'}}(3^{k'} \mid \mathbf{x}_i^{k'}) + p_f^{k'} \right) \cdots d\mu(\mathbf{x}^{1:k-1}) \right). \end{aligned} \quad (82)$$

Noting that the summand in (82) has no dependence on $p_f^{k'}$, the dominated convergence theorem [60] may be trivially applied to the constant $p_f^{k'}$ term to give

$$\begin{aligned} \lim_{p_f^{k'} \rightarrow 0^+} p_{X^k|Z^{1:k}}(\mathbf{x} \mid \mathfrak{Z}^{1:k}) &= \\ \left(\cdots \int_{\mathbb{S}} \mathbf{1}_{A^{k'}}(\mathbf{x}) \left(\sum_{i \in I} p_{d,i}^{k'} f_{Z^{k'}|X_S^{k'}}(3^{k'} \mid \mathbf{x}_i^{k'}) \right. \right. & \\ \left. \left. + \lim_{p_f^{k'} \rightarrow 0^+} p_f^{k'} \right) \cdots d\mu(\mathbf{x}^{1:k-1}) \right) & \\ = \left(\cdots \int_{\mathbb{S}} \mathbf{1}_{A^{k'}}(\mathbf{x}) \left(\sum_{i \in I} p_{d,i}^{k'} f_{Z^{k'}|X_S^{k'}}(3^{k'} \mid \mathbf{x}_i^{k'}) \right. \right. & \\ \left. \left. + 0 \right) \cdots d\mu(\mathbf{x}^{1:k-1}) \right), & \end{aligned} \quad (83)$$

and the relation

$$\lim_{p_f^{k'} \rightarrow 0^+} p_{X^k|Z^{1:k}}(\mathbf{x} \mid \mathfrak{Z}^{1:k}) = p_{X^k|Z^{1:k}}(\mathbf{x} \mid \mathfrak{Z}^{1:k})|_{p_f^{k'}=0} \quad (84)$$

thus holds. Equation (64) may now be written as

$$\begin{aligned} \mathbf{P}_{t,i}^k|_{p_f^{k'}=0} &= \int_{r_i^k} p_{X_i^k|Z^{1:k}}(\mathbf{x}_i \mid \mathfrak{Z}^{1:k})|_{p_f^{k'}=0} d\mu(\mathbf{x}_i) \\ &= \int_{r_i^k} \lim_{p_f^{k'} \rightarrow 0^+} p_{X_i^k|Z^{1:k}}(\mathbf{x}_i \mid \mathfrak{Z}^{1:k}) d\mu(\mathbf{x}_i). \end{aligned} \quad (85)$$

As the $p_{X_i^k|Z^{1:k}}(\mathbf{x}_i \mid \mathfrak{Z}^{1:k})$ may be assumed bounded, and because the limit in (84) is satisfied pointwise, the dominated convergence theorem may be applied a second time to yield

$$\mathbf{P}_{t,i}^k|_{p_f^{k'}=0} = \lim_{p_f^{k'} \rightarrow 0^+} \int_{r_i^k} p_{X_i^k|Z^{1:k}}(\mathbf{x}_i \mid \mathfrak{Z}^{1:k}) d\mu(\mathbf{x}_i). \quad (86)$$

By way of the algebraic and integration steps between (64) and (78), the permanent ratio becomes

$$\mathbf{P}_{t,i}^k|_{p_f^{k'}=0} = \lim_{p_f^{k'} \rightarrow 0^+} \mathbf{M}_{t,i}^k \frac{\text{Per}(\mathbf{M}^k(t;i))}{\text{Per}(\mathbf{M}^k)}. \quad (87)$$

Provided that $\lim_{p_f^{k'} \rightarrow 0^+} \mathbf{M}^k$ possesses at least one non-zero permutation, the denominator is positive, and the limit of (87) may be distributed under the permanent operations through successive applications of the algebraic limit theorem [61]

$$\mathbf{P}_{t,i}^k|_{p_f^{k'}=0} = \left(\lim_{p_f^{k'} \rightarrow 0^+} \mathbf{M}_{t,i}^k \right) \frac{\text{Per}(\lim_{p_f^{k'} \rightarrow 0^+} \mathbf{M}^k(t;i))}{\text{Per}(\lim_{p_f^{k'} \rightarrow 0^+} \mathbf{M}^k)}. \quad (88)$$

The limiting process thus amounts to calculating

$$\begin{aligned} \lim_{p_i^k \rightarrow 0^+} \mathbf{M}^k &= \left[\lim_{p_i^k \rightarrow 0^+} \mathbf{A}^k \mid \lim_{p_i^k \rightarrow 0^+} \mathbf{B}^k \mathbf{J} \right] \\ &= \left[\lim_{p_i^k \rightarrow 0^+} \mathbf{A}^k \mid \mathbf{B}^k \mathbf{J} \Big|_{p_i^k=0} \right], \end{aligned} \quad (89)$$

where the second equality results from the fact that $\mathbf{B}^k \mathbf{J}$ submatrix contains only linear terms of p_i^k . The remaining limit in (89) may be found by calculating the limits to the individual $a_{t,i}^k$. Noting the definition of $\mathbf{o}(\cdot)$ in Condition 5 in §2-B and denoting its preimage by $\mathbf{o}^{-1}(\cdot)$, the recursion of (27) gives

$$\lim_{p_i^k \rightarrow 0^+} a_{t,i}^k = \lim_{p_i^k \rightarrow 0^+} a_{t,i}^1 \prod_{\substack{k'' \in \mathbf{o}^{-1}(t) \\ k'' \leq k}} K_i^{k''}, \quad (90)$$

which, using (29), may be expanded as

$$\begin{aligned} \lim_{p_i^k \rightarrow 0^+} a_{t,i}^k &= \\ \lim_{p_i^k \rightarrow 0^+} a_{t,i}^1 &\prod_{\substack{k'' \in \mathbf{o}^{-1}(t) \\ k'' \leq k}} \left(\int_{\mathcal{S}^2} g_{i,3^{k''}}(\mathbf{x}_i'') \right. \\ &\cdot \left. f_{X_i^{k''} | X_i^{k''-1}}(\mathbf{x}_i'' | \mathbf{x}_i'') p_{t,i}^{k''-1}(\mathbf{x}_i'') d\mu(\mathbf{x}_i'') d\mu(\mathbf{x}_i'') \right). \end{aligned} \quad (91)$$

The $g_{i,3^{k''}}$ may be substituted with the right-hand side of (26), and after applying a process based on the steps of (81)–(84), equation (91) simplifies to

$$a_{t,i}^k \Big|_{p_i^k=0} = a_{t,i}^1 \prod_{\substack{k'' \in \mathbf{o}^{-1}(t) \\ k'' \leq k}} K_i^{k''} \Big|_{p_i^k=0}. \quad (92)$$

Finally,

$$\mathbf{P}_{t,i}^k \Big|_{p_i^k=0} = (\mathbf{M}_{t,i}^k \Big|_{p_i^k=0}) \frac{\text{Per}(\mathbf{M}^k(t; i) \Big|_{p_i^k=0})}{\text{Per}(\mathbf{M}^k \Big|_{p_i^k=0})}, \quad (93)$$

allowing p_i^k to be set to zero in (26) and (27).

A zero row \mathbf{B}_i^k signifies that at least one measurement with a zero false-alarm probability was encountered over the history of track t . The target corresponding to such a track is therefore certain to exist, and consequently, all track-identity permutations of non-zero weight include this track. Therefore, the \mathbf{B}^k matrix may be seen as a tabulation of evidence supporting the existence of individual tracks by maintaining running products of track-level false alarm probabilities.

The denominator of (93) may become zero over the course of performing the limiting steps on the series of vanishing p_i^k , a condition that arises by an over-designation of zero false-alarm probabilities that renders the number of certain tracks larger than the number of identities. This scenario occurs when one or more measurement likelihood functions are defined in a manner that is incongruent with the properties of the

statistical system. While the difficulty associated with undefined denominators may be superficially remedied by altogether avoiding zero false-alarm probabilities, a preferable solution entails addressing the underlying deficiencies in the affected measurement functions.

APPENDIX C RYSER FORMULA FOR MATRICES WITH DUPLICATE ROWS AND COLUMNS

Noting that members of the power set $Y \in \mathcal{P}(\mathbb{N}_n)$ may be brought into one-one correspondence with those of the set of n -dimensional 0-1 vectors ($\mathbf{c}_Y \in 2^n \equiv \prod_n \{0, 1\}$), (35) may be rewritten as

$$\text{Per}(\mathbf{M}) = (-1)^n \sum_{\mathbf{c}_Y \in 2^n} (-1)^{\|\mathbf{c}_Y\|_1} \prod_{q=1}^n \sum_{y \in Y} \mathbf{M}_{q,y}, \quad (94)$$

where $\|\mathbf{c}_Y\|_1$ is the L^1 norm of \mathbf{c}_Y (the sum of the entries in \mathbf{c}_Y), which is equivalent to the cardinality of Y . A further simplification results by expressing the trailing sum as a dot product

$$\text{Per}(\mathbf{M}) = (-1)^n \sum_{\mathbf{c}_Y \in 2^n} (-1)^{\|\mathbf{c}_Y\|_1} \prod_{q=1}^n \mathbf{c}_Y \cdot \mathbf{M}_q, \quad (95)$$

where \mathbf{M}_q is the q th row of \mathbf{M} . In general, the set 2^n may be decomposed as the Cartesian product

$$2^n = 2^{n_1} \times 2^{n_2} \times \dots \times 2^{n_{\hat{n}}} \quad (96)$$

for any set of positive integers n_j that satisfies $n = \sum_{j=1}^{\hat{n}} n_j$. Under such a decomposition, $\|\mathbf{c}_Y\|_1 = \|\mathbf{c}_1\|_1 + \|\mathbf{c}_2\|_1 + \dots + \|\mathbf{c}_{\hat{n}}\|_1$, and $\mathbf{c}_Y = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{\hat{n}})$. Consequently, (95) becomes

$$\begin{aligned} \text{Per}(\mathbf{M}) &= (-1)^n \sum_{\mathbf{c}_1 \in 2^{n_1}} \sum_{\mathbf{c}_2 \in 2^{n_2}} \dots \sum_{\mathbf{c}_{\hat{n}} \in 2^{n_{\hat{n}}}} \\ &(-1)^{\|\mathbf{c}_1\|_1 + \|\mathbf{c}_2\|_1 + \dots + \|\mathbf{c}_{\hat{n}}\|_1} \prod_{q=1}^n \mathbf{c}_Y \cdot \mathbf{M}_q. \end{aligned} \quad (97)$$

Each c_j may be associated with a distinct group of repeated columns in \mathbf{M} , provided that each group is formed by a single, contiguous submatrix.¹⁸ In this case, the j th group of unique columns interacts only with \mathbf{c}_j in the dot product of (97). By assigning the number of column groups in \mathbf{M} to \hat{n} and setting n_j to the multiplicity of the j th group, the permanent may be written as

$$\begin{aligned} \text{Per}(\mathbf{M}) &= (-1)^n \sum_{\mathbf{c}_1 \in 2^{n_1}} \sum_{\mathbf{c}_2 \in 2^{n_2}} \dots \sum_{\mathbf{c}_{\hat{n}} \in 2^{n_{\hat{n}}}} \\ &(-1)^{\|\mathbf{c}_1\|_1 + \|\mathbf{c}_2\|_1 + \dots + \|\mathbf{c}_{\hat{n}}\|_1} \\ &\cdot \prod_{q=1}^n (\|\mathbf{c}_1\|_1, \|\mathbf{c}_2\|_1, \dots, \|\mathbf{c}_{\hat{n}}\|_1) \cdot \bar{\mathbf{M}}'_q, \end{aligned} \quad (98)$$

¹⁸This implies that each column belonging to the j th column group appears to the left of every column in the $(j+1)$ th group. A matrix can always be brought into this form, as the permanent is invariant under column permutations.

where $\bar{\mathbf{M}}'_q$ is the q th row of the smaller matrix $\bar{\mathbf{M}}'$, which comprises the unique columns of \mathbf{M} . As the summands in (98) depend only on the L^1 norm of the \mathbf{c}_j , the sum may be rewritten as

$$\begin{aligned} \text{Per}(\mathbf{M}) &= (-1)^n \sum_{c_1=1}^{n_1} \binom{c_1}{n_1} \sum_{c_2=1}^{n_2} \binom{c_2}{n_2} \cdots \sum_{c_{\hat{n}}=1}^{n_{\hat{n}}} \binom{c_{\hat{n}}}{n_{\hat{n}}} \\ &\quad \cdot (-1)^{c_1+c_2+\dots+c_{\hat{n}}} \prod_{q=1}^n (c_1, c_2, \dots, c_{\hat{n}}) \cdot \bar{\mathbf{M}}'_q, \end{aligned} \quad (99)$$

where the binomial coefficient $\binom{c_j}{n_j}$ gives the number of \mathbf{c}_j vectors with an L^1 norm of c_j . Finally, the sums and binomial coefficients may be collected by defining $\mathbf{d} = (c_1, c_2, \dots, c_{\hat{n}})$ to give

$$\text{Per}(\mathbf{M}) = \sum_{\mathbf{d} \in \prod \mathbb{N}_{n_j}} (-1)^{n+\|\mathbf{d}\|_1} \left[\prod_{j=1}^{\hat{n}} \binom{d_j}{n_j} \right] \prod_{q=1}^n \mathbf{d} \cdot \bar{\mathbf{M}}'_q. \quad (100)$$

Inspection of (99)—which is functionally equivalent to (100)—reveals this reduced form to have a more favourable complexity of $\mathcal{O}(n \prod_{j=1}^{\hat{n}} (n_j + 1))$ when the sum are evaluated using a generalized Gray code (i.e., consecutive summation terms differ by only a single d_j in \mathbf{d} such that $(d_j + 1 \bmod n_j)$ yields the d_j of the subsequent term).

Finally, when the matrix possess repeated rows, the dot products $\mathbf{d} \cdot \bar{\mathbf{M}}'_{q_1}$ and $\mathbf{d} \cdot \bar{\mathbf{M}}'_{q_2}$ are equal when q_1 and q_2 belong to a common row group. The identical terms may be collected and replaced with a single term exponentiated to the size of the row group. Equation (100) then becomes

$$\begin{aligned} \text{Per}(\mathbf{M}) &= \sum_{\mathbf{d} \in \prod \mathbb{N}_{n_j}} (-1)^{n+\|\mathbf{d}\|_1} \\ &\quad \cdot \left[\prod_{j=1}^{\hat{n}} \binom{d_j}{n_j} \right] \prod_{q=1}^{\hat{m}} (\mathbf{d} \cdot \bar{\mathbf{M}}_q)^{m_q}. \end{aligned} \quad (101)$$

where \hat{m} is the number of unique rows, m_q is the size of the q th group of duplicate rows, and $\bar{\mathbf{M}}_q$ is the q th row of matrix $\bar{\mathbf{M}}$ comprising only unique rows (and columns). The time complexity of this simplification is $\mathcal{O}(\hat{m} \prod_j (n_j + 1))$. Finally, as the permanent is invariant under matrix transposition, the minimum complexity becomes equation (37).

REFERENCES

- [1] S. Challa, M. R. Morelande, D. Mušicki, and R. J. Evans *Fundamentals of Object Tracking*. Cambridge: Cambridge University Press, 2011.
- [2] S. Challa, R. J. Evans, and D. Musicki *Target Tracking—A Bayesian Perspective*, in *Fourteenth International Conference on Digital Signal Processing*, vol. 1. IEEE, 2002, pp. 437–440.

- [3] I. R. Goodman, R. P. Mahler, and H. T. Nguyen *Mathematics of Data Fusion*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1997.
- [4] R. E. Kalman *A New Approach to Linear Filtering and Prediction Problems*, *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
- [5] C. K. Chui and G. Chen *Kalman Filtering: With Real-Time Applications*. Springer, 2009.
- [6] S. S. Blackman *Multiple-Target Tracking with Radar Applications*. Dedham, MA: Artech House, 1986.
- [7] M. Mallick, V. Krishnamurthy, and B.-N. Vo *Integrated Tracking, Classification, and Sensor Management: Theory and Applications*. Wiley-IEEE Press, 2012.
- [8] R. P. Mahler *Statistical Multisource-Multitarget Information Fusion*. Norwood, MA: Artech House, 2007.
- [9] B. Ristic, N. Gordon, and A. Bessell *On Target Classification using Kinematic Data*, *Information Fusion*, vol. 5, no. 1, pp. 15–21, 2004.
- [10] M. L. Krieg *Joint Multi-Sensor Kinematic and Attribute Tracking using Bayesian Belief Networks*, in *Proceedings of the 2003 International Conference on Information Fusion*, 2003, pp. 17–24.
- [11] ——— *“Kinematic and Attribute Fusion Using a Bayesian Belief Network Framework,”* Defence Science and Technology Organisation, PO Box 1500 Edinburgh, South Australia 5111, Australia, Tech. Rep., 2006.
- [12] W. Mei, G.-l. Shan, and Y.-f. Wang *A second-order uncertainty model for target classification using kinematic data*, *Information Fusion*, vol. 12, no. 2, pp. 105–110, 2011.
- [13] D. Angelova and L. Mihaylova *Joint Target Tracking and Classification with Particle Filtering and Mixture Kalman Filtering using Kinematic Radar Information*, *Digital Signal Processing*, vol. 16, no. 2, pp. 180–204, 2006.
- [14] F. Palmieri and D. Ciuonzo *Data Fusion with Entropic Priors*, *Proceedings of 20th Workshop on Neural Networks, WIRN, Frontiers in Artificial Intelligence and Applications*, vol. 226, pp. 107–114, 2011.
- [15] J. Teti Jr, R. Gorman, and W. Berger *A Multifeature Decision Space Approach to Radar Target Identification*, *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 32, no. 1, pp. 480–487, 1996.
- [16] S. Challa and G. W. Pulford *Joint Target Tracking and Classification Using Radar and ESM Sensors*, *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 3, pp. 1039–1055, 2001.
- [17] B. T. Vo and B. N. Vo *Tracking, Identification, and Classification with Random Finite Sets*, in *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2013, pp. 87450D–87450D.

- [18] J. Shin, N. Lee, S. Thrun, and L. Guibas
Lazy Inference on Object Identities in Wireless Sensor Networks,
in *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE Press, 2005, p. 23.
- [19] B. Schumitsch, S. Thrun, G. Bradski, and K. Olukotun
The Information-Form Data Association Filter,
in *Advances in Neural Information Processing Systems*, 2005, pp. 1193–1200.
- [20] J. Shin, L. J. Guibas, and F. Zhao
A Distributed Algorithm for Managing Multi-Target Identities in Wireless Ad-Hoc Sensor Networks,
in *Information Processing in Sensor Networks*. Springer, 2003, pp. 223–238.
- [21] H. Balakrishnan, I. Hwang, and C. J. Tomlin
Polynomial Approximation Algorithms for Belief Matrix Maintenance in Identity Management,
in *43rd IEEE Conference on Decision and Control*, vol. 5. IEEE, 2004, pp. 4874–4879.
- [22] I. Hwang, K. Roy, H. Balakrishnan, and C. Tomlin
A Distributed Multiple-Target Identity Management Algorithm in Sensor Networks,
in *43rd IEEE Conference on Decision and Control*, vol. 1. IEEE, 2004, pp. 728–734.
- [23] B. Schumitsch, S. Thrun, L. J. Guibas, and K. Olukotun
The Identity Management Kalman Filter (IMKF),
in *Robotics: Science and Systems*. Citeseer, 2006.
- [24] J. Huang, C. Guestrin, and L. Guibas
Fourier theoretic probabilistic inference over permutations,
Journal of Machine Learning Research, vol. 10, pp. 997–1070, 2009.
- [25] ———
Efficient Inference for Distributions on Permutations,
in *Advances in Neural Information Processing Systems*, 2007, pp. 697–704.
- [26] R. Kondor, A. Howard, and T. Jebara
Multi-Object Tracking with Representations of the Symmetric Group,
in *International Conference on Artificial Intelligence and Statistics*, 2007, pp. 211–218.
- [27] A. S. Asratian
Bipartite Graphs and their Applications.
Cambridge University Press, 1998, no. 131.
- [28] R. E. Burkard, M. Dell’Amico, and S. Martello
Assignment Problems, Revised Reprint.
Philadelphia: Society for Industrial and Applied Mathematics, 2009.
- [29] A. B. Badiru and O. A. Omitaomu
Computational Economic Analysis for Engineering and Industry.
Boca Raton, FL: CRC Press, 2010.
- [30] M. Fujita and J.-F. Thisse
Economics of Agglomeration: Cities, Industrial Location, and Globalization.
New York: Cambridge University Press, 2013.
- [31] C. K. Mustafi
Operations Research methods and practice.
New Age International, 1996.
- [32] H. S. Kasana and K. D. Kumar
Introductory Operations Research: Theory and Applications.
Berlin Heidelberg: Springer-Verlag, 2004.
- [33] J. Collins and J. Uhlmann
Efficient Gating in Data Association with Multivariate Gaussian Distributed States,
IEEE Transactions on Aerospace and Electronic Systems, vol. 28, no. 3, pp. 909–916, July 1992.
- [34] M. Jerrum, A. Sinclair, and E. Vigoda
A Polynomial-Time Approximation Algorithm for the Permanent of a Matrix with Non-Negative Entries,
in *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*. ACM, 2001, pp. 712–721.
- [35] ———
A Polynomial-Time Approximation Algorithm for the Permanent of a Matrix with Nonnegative Entries,
Journal of the ACM (JACM), vol. 51, no. 4, pp. 671–697, 2004.
- [36] I. Bezáková, D. Štefankovic, V. V. Vazirani, and E. Vigoda
“Approximating the Permanent in $O(n^7)$ Time,”
2004.
- [37] E. W. Weisstein
CRC Concise Encyclopedia of Mathematics.
Boca Raton, FL: CRC Press, 2003.
- [38] L. Hogben
Handbook of Linear Algebra.
Boca Raton, FL: Chapman and Hall/CRC, 2006.
- [39] O. Goldreich
Computational Complexity: A Conceptual Perspective.
New York: Cambridge University Press, 2008.
- [40] T. Fortmann and Y. Bar-Shalom
Tracking and Data Association.
San Diego: Academic Press, 1988.
- [41] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell
Bayesian Multiple Target Tracking.
Norwood, MA: Artech House, 2013.
- [42] T. Granlund
GNU MP, 5th ed., Free Software Foundation, 2013.
[Online]. Available: <https://gmplib.org/manual/>.
- [43] S. Friedland
A Lower Bound for the Permanent of a Doubly Stochastic Matrix,
Annals of Mathematics, pp. 167–176, 1979.
- [44] Lindley, Dennis V
Understanding Uncertainty.
Hoboken, NJ: John Wiley & Sons, 2006.
- [45] A. W. Marshall and I. Olkin
Scaling of Matrices to Achieve Specified Row and Column Sums,
Numerische Mathematik, vol. 12, no. 1, pp. 83–90, 1968.
- [46] R. Sinkhorn
A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices,
The Annals of Mathematical Statistics, vol. 35, no. 2, pp. 876–879, 1964.
- [47] M. Fürer
Faster Integer Multiplication,
SIAM Journal on Computing, vol. 39, no. 3, pp. 979–1005, 2009.
- [48] B. Kalantari and L. Khachiyan
On the Complexity of Nonnegative-Matrix Scaling,
Linear Algebra and its Applications, vol. 240, pp. 87–103, 1996.
- [49] L. G. Valiant
The Complexity of Computing the Permanent,
Theoretical Computer Science, vol. 8, no. 2, pp. 189–201, 1979.
- [50] I. Bezáková, D. Štefankovič, V. V. Vazirani, and E. Vigoda
Accelerating Simulated Annealing for the Permanent and Combinatorial Counting Problems,
in *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM, 2006, pp. 900–907.
- [51] I. Bezáková, D. Štefankovic, V. V. Vazirani, and E. Vigoda
Accelerating Simulated Annealing for the Permanent and Combinatorial Counting Problems,
SIAM Journal on Computing, vol. 37, no. 5, pp. 1429–1454, 2008.

- [52] M. Huber and J. Law
Fast Approximation of the Permanent for Very Dense Problems,
in *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2008, pp. 681–689.
- [53] B. Huang and T. Jebara
“Approximating the Permanent with Belief Propagation,”
arXiv preprint arXiv:0908.1769, 2009.
- [54] P. O. Vontobel
The Bethe Permanent of a Non-Negative Matrix,
in *48th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2010, pp. 341–346.
- [55] Y. Watanabe and M. Chertkov
Belief Propagation and Loop Calculus for the Permanent of a Non-Negative Matrix,
Journal of Physics A: Mathematical and Theoretical, vol. 43, no. 24, p. 242002, 2010.
- [56] J. L. Williams and R. A. Lau
“Approximate Evaluation of Marginal Association Probabilities with Belief Propagation,”
arXiv preprint arXiv:1209.6299, 2012.
- [57] M. Chertkov and A. B. Yedidia
“Computing the Permanent with Belief Propagation,”
arXiv preprint arXiv:1108.0065, 2011.
- [58] Y. Bar-Shalom and X. R. Li
Estimation and Tracking: Principles, Techniques, and Software.
Boston: Artech House, 1993.
- [59] K. L. Stern
“Hungarian Method C++ Source Code,”
Online, 2012. [Online]. Available: <https://github.com/KevinStern/software-and-algorithms-cpp/blob/master/src/hungarian.h>.
- [60] M. Capinski and P. E. Kopp
Measure, Integral and Probability.
London: Springer, 2004.
- [61] F. L arsson
Lectures on Real Analysis.
New York: Cambridge University Press, 2012, vol. 21.



Dominic Schaub received his B.Sc. in Computer Engineering and Ph.D. in Electrical Engineering from the University of Manitoba in 2004 and 2011, respectively. In 2011 he joined Defence Research and Development Canada, Atlantic Research Centre.

Cognitive Video Streaming

D. PASUPULETI
 P. MANNARU
 B. BALASINGAM
 M. BAUM
 K. PATTIPATI
 P. WILLETT
 C. LINTZ
 G. COMMEAU
 F. DORIGO
 J. FAHRNY

Video-on-demand (VoD) streaming services are becoming increasingly popular due to their flexibility in allowing users to access their favorite video content anytime and anywhere from a wide range of access devices, such as smart phones, computers and TV. The content providers rely on highly satisfied subscribers for revenue generation and there have been significant efforts in developing approaches to “estimate” the quality of experience (QoE) of VoD subscribers. However, a key issue is that QoE can be difficult to measure directly from residential and mobile user interactions with content. Hence, appropriate proxies need to be found for QoE, via the streaming metrics (the QoS metrics) that are largely based on initial startup time, buffering delays, average bit rate and average throughput and other relevant factors such as the video content and user behavior and other external factors. The ultimate objective of the content provider is to elevate the QoE of all the subscribers at the cost of minimal network resources, such as hardware resources and bandwidth.

In this paper, first, we propose a *cognitive video streaming strategy* in order to ensure the QoE of subscribers, while utilizing minimal network resources. The proposed cognitive video streaming architecture consists of an *estimation module*, a *prediction module*, and an *adaptation module*. Then, we demonstrate the prediction module of the cognitive video streaming architecture through a *play time prediction tool*. For this purpose, the applicability of different machine learning algorithms, such as the k-nearest neighbor, neural network regression, and survival models are experimented with; then, we develop an approach to identify the most relevant factors that contributed to the prediction. The proposed approaches are tested on dataset provided by Comcast Cable.

Manuscript received June 10, 2016; revised June 24, 2016; released for publication August 10, 2016.

Refereeing of this contribution was handled by Benjamin Slocumb.

Authors’ addresses: D. Pasupuleti, P. Mannaru, B. Balasingam, M. Baum, K. Pattipati, and P. Willett are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, 06269, USA (E-mail: {devaki, pujitha.mannaru, bala, mbaum, krishna, willett}@engr.uconn.edu). C. Lintz, G. Commeau, F. Dorigo, and J. Fahrny are with Comcast Corporation, USA (E-mail: {Christopher.Lintz, Gabriel.Commeau, francesco.dorigo, Jim.Fahrny@cable.comcast.com}).

Dr. Balasingam is the corresponding author.

Some initial works have been published in [40] and [41].

1557-6418/17/\$17.00 © 2017 JAIF

I. INTRODUCTION

Major advances in wireless communication and consumer electronics of the past decade have disrupted the traditional ways in which people used to consume video programs. In a traditional setting (see Figure 1), a viewer has to “tune-in” to a TV station via cable, satellite or on-air receiver in order to watch or record his/her favorite program. Today, with internet and wireless broadband connectivity, there are several options for a viewer to watch his/her favorite programs at the time of his/her convenience using a device of his/her choice (see Figure 2), such as a smart phone, tablet, computer or TV. As a result, the video distribution strategy also has gone through major changes.

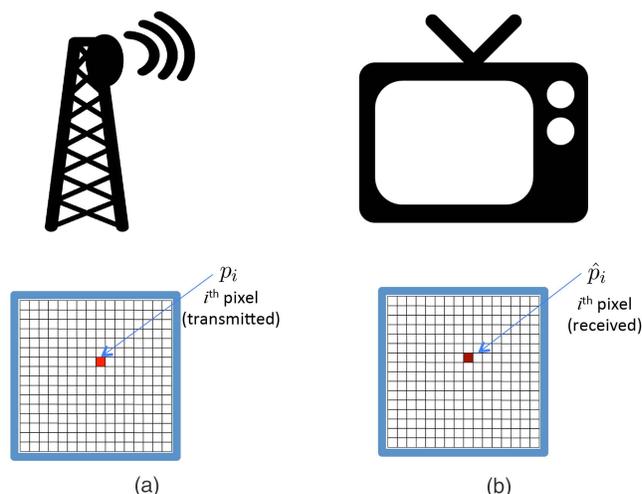


Fig. 1. **Traditional video transmission and reception.** Traditional QoS metrics try to quantify viewers’ perception using objective metrics computed based on transmitted and received frame sequences. (a) Video transmission. (b) Video reception.

A brief description of each of the blocks in Figure 2 is given below:

- **Content.** Content can be divided into online streaming, i.e., regular TV programs, and recorded programs that are delivered as video-on-demand (VoD), the focus of this paper. In VoD, a viewer browses through the lists of available videos and selects one to play. Unlike online streaming, VoD offers the capability to pause and resume videos at any time.
- **Delivery service.** Delivery service providers, such as cable networks, bring the videos to the viewers. Usually, the viewer has to be a subscriber to the delivery service provider in order to get access to the content.
- **Viewer.** The viewer accesses the videos using devices, such as smart phones, tablets, TV and Computer. Each viewing device may have different connectivity and bandwidth. Depending on the access device (portable or desktop), the characteristics of the viewer might be different as well. For example, a viewer may be willing to tolerate intermittent buffering events and

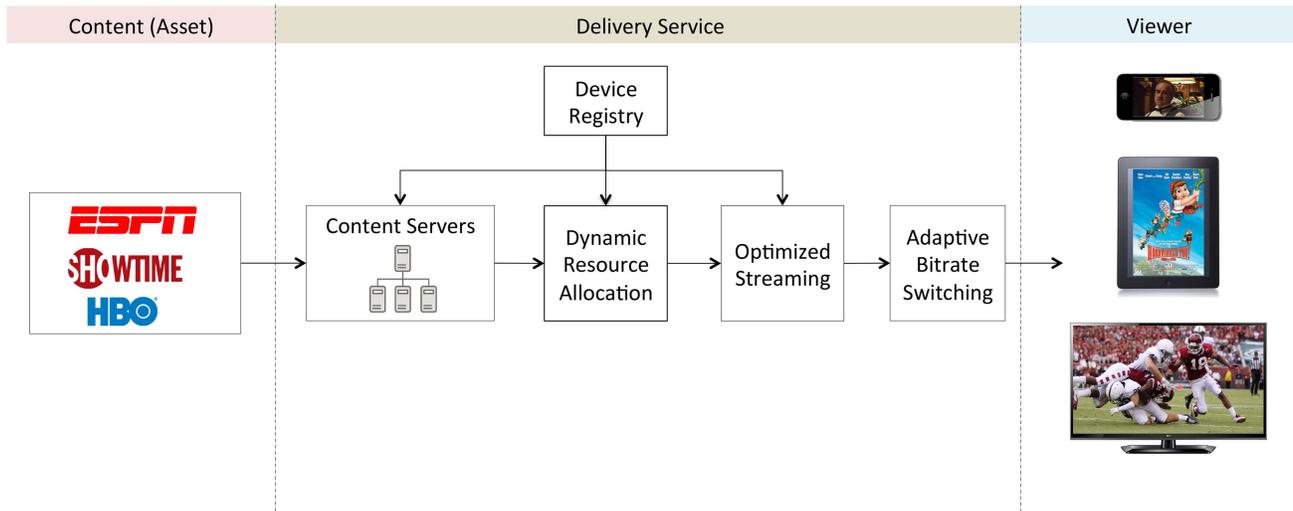


Fig. 2. **Description of a video-on-demand (VoD) system.** Unlike traditional video transmission systems, the viewers have the option of choosing from a large amount of video content or to select watching online video streaming.

longer startup times in a smart phone, while exhibiting lesser tolerance towards similar events in a TV.

- *Content servers.* Content servers respond to the VoD requests and stream videos to the viewers. Based on the popularity of particular videos, content servers adjust content delivery priorities in order to provide good QoS to the viewers.
- *Dynamic resource allocation.* Content service providers respond to rapidly increasing/decreasing demands to particular videos, anticipated and unexpected, such as major sports events and unexpected world events, by dynamically adjusting the streaming capacity of videos.
- *Optimized streaming.* Optimized streaming algorithms aim to deliver high quality videos at reduced cost (bandwidth) to the viewer. This is achieved by efficiently compressing subsequent video frames. Some other constraints include the power and memory requirements of the video player at the viewing devices.
- *Device registry.* An important challenge in maintaining superior quality of online video streaming is the increasing number of different types of devices available to viewers in order to play videos. Each of these devices has different hardware and software capabilities. Knowing the exact capabilities of a particular device is important in optimizing the video streaming.
- *View logs.* These represent feedback data from the video players to the content delivery service providers. The feedback contains data, such as bit rate, buffering information and media-failed events that are useful in assessing the quality of experience of the viewer.
- *Adaptive bitrate switching.* In mobile video devices, the available bandwidth can vary depending on the location of the receiver. For example, moving the device (e.g., moving between different parts of a house, traveling in a vehicle, walking through a mall, etc.), can result in varying download bandwidths at

the device. The video streaming algorithms respond to this by adjusting the bit-rate of the content.

The quality of user experience has been a concern in both traditional and the emerging content delivery systems. In the traditional video broadcasting scenario, the issue of video quality arises due to video transmission and processing manifested in the form of noise, jitter, shape transformation, and so on. Traditional QoS assessment schemes focused on quantifying the perception of the viewers on videos with varying types and degrees of video transmission distortions; such distortions are generally defined as the QoS metrics, such as peak signal to noise ratio (PSNR,[50]), video quality metric (VQM,[42]), moving picture quality metric (MPQM,[48]), structural similarity index (SSIM,[51]), and noise quality measure (NQM,[14]). The viewers' perceptions as a result of varying QoS are obtained through subjective methods and quantified usually as a mean opinion score (MOS,[23]). The MOS, scaled between 0 and 5, represents the perceptual quality of the video; very pleasant and clear viewing experience will result in MOS of 5 and an intolerable video will have an MOS of 0. When poor QoS is detected in some areas, the broadcasters must find ways to increase the signal to noise ratio to the affected area; this can be achieved by increasing the power of existing transmitters or by installing additional transmitters (or repeaters) in the affected area. The MOS scheme in traditional TV broadcasting enjoys wide acceptance (see [20]).

In VoD, the QoS factors are different from those in traditional video; some widely used QoS factors are based on startup time, buffering and transmission bitrate. The startup time is defined as the time between the initial video request (such as clicking on the play button on a web interface) and the time of playing the first video frame on the screen. Higher startup time can cause the viewer to abandon the video [26]. There

are several factors affecting the startup time; connection bandwidth of the viewer, capability of the video distribution server, and network delays are a few of them. In order to reduce the startup time, the player “buffers” a portion of the video before it starts and the rest of the video is continuously buffered while the video is still playing. Buffering is supposed to happen in the background while the video is playing; however, similar to startup time, non ideal streaming conditions cause the player to pause and wait for the data to be buffered. It is reported in [26] that buffering delays negatively impact the likelihood of a viewer’s return to the content provider. Adaptive bitrate switching [28] allows content providers to reduce the startup and buffering delays by adaptively switching the frame quality of the video based on the bandwidth and other hardware capability of the video player. The higher the bandwidth and processing capabilities of the player, the higher the bit-rate and quality of the video; the bitrate serves as a QoS factor. High average bitrate over a certain period of time indicates that the rendering quality was high and vice versa; frequent bitrate switching with high variation indicates poor quality of experience due to volatile bandwidth. Analysis of viewer responses to the startup time, buffering and bitrate related QoS factors are reported in [15]. The adaptive bitrate streaming technique has been widely adopted by many existing content providers; in [39] and [24], a general overview of the widely adopted HTTP adaptive streaming (HAS) protocol is provided.

Adaptive video streaming itself is challenging and diverse approaches have been published in the literature [45]. Most of the adaptive streaming strategies recommend adapting the bitrate based on buffering events [17]. Other than adaptive streaming, there are several suggestions in the literature to enhance a specific aspect of QoE; in [4], an approach is suggested to enhance the accessibility in shared video forums; [5] suggests exploiting the knowledge that concurrent viewers are viewing a specific content and using peer-to-peer (P2P) strategies to offload some of the workload of the content servers; an approach for client side server selection is presented in [29]; in [44], the QoE is modeled based on a packet loss model; in [49], the QoE is modeled in terms of the QoS factors such as loss, delay and jitter; and [11] talks about providing good quality video, while being aware of the bandwidth quota of the user.

Current adaptive streaming and other approaches developed to enhance QoE are designed to “react” to the QoS factors (that are largely based on startup time, buffer level and average bitrate) from the viewer’s device. This does not guarantee that the quality of experience (QoE) of the viewer will be improved as a result. For example, the decision to downgrade the bitrate (i.e, the quality of the video) as a result of buffering delay may not be appreciated by all viewers; to make things worse, the same viewer might have varying preferences depending on circumstances such as the time of day. Further, there is explosive growth in the internet

traffic caused by videos delivered by content delivery networks; this trend is expected to continue as more and more viewers turn from traditional TV to VoD [1]. Expanding the network infrastructure is costly and time consuming; a QoE based adaptive streaming will help ease some of the strain on the network by increasing the bitrate only when it is likely to advance the QoE of the viewer. In other words, a better and futuristic adaptive streaming technique has to be “proactive” rather than reactive.

The first step in QoE-based adaptive video streaming is to come up with accurate methods of estimating the QoE of the viewer. Taking cues from the widely adopted MOS in traditional TV, some initial attempts were made in [36] to estimate the MOS in response to the QoS factors of VoD. However, unlike traditional video, the MOS obtained through a limited experiment is unable to represent the viewers’ perception in a wide ranging VoD scenario. It is found that the viewers react differently to the same video content with the same QoS factor; viewers seemed to tolerate QoS deficiencies in live video compared to non-live content [7]; viewers from well connected devices (those with better connection bandwidth) are found to be less tolerant compared to their low-bandwidth counterparts.

A VoD viewer has millions and millions of videos to choose from. Instead of traditional TV, there are devices of convenience (with trade offs) for a particular time of day; video in a smart phone might come with too many buffering events and blurry images compared to a TV; however, its portability is appealing to a certain viewer during day-time; the same viewer might prefer to continue the same video using TV during the evening. For content providers, the objective has become one of attracting and retaining subscribers by providing superior quality of experience. Due to the nature of VoD consumption, it is impossible to capture the QoE in terms of a single metric, such as MOS. Hence the MOS, which is subjectively estimated using a particular viewing scenario, is not adequate to quantify viewers’ QoE [10].

Recently, there have been attempts to estimate QoE from user data; these approaches are generally termed “passive,” “online” or “indirect” approaches of estimating QoE. In [6], [7], it was suggested to create a predictive model of viewer engagement (such as total play time, number of visits and probability of return) based on the observed QoS factors. A machine learning framework to estimate the QoE in mobile applications was proposed in [3]; this approach requires training data from past “good QoE” and “poor QoE” instances. Table I gives a comparative summary of existing QoS literature corresponding to traditional video transmissions and QoE metrics corresponding to VoD and internet video.

The existing approaches focus heavily on modeling the QoE as related to the QoS factors only. However, even though the QoE is significantly influenced by the

TABLE I
Summary of QoE Approaches in Traditional TV and VoD

	Traditional Video	VoD
QoS factors	<ul style="list-style-type: none"> • PSNR—Peak Signal to Noise Ratio [50] • VQM—Video Quality Metric [42] • MPQM—Moving Pictures Quality Metric [48] • SSIM—Structural Similarity Index [51] • NQM—Noise Quality Measure [14] 	<ul style="list-style-type: none"> • Startup time [15] • Buffering time [43] • Buffering count [43] • Buffering ratio [15] • Rate of buffering events [15] • Normalized re-buffer delay [25] • Average bit rate [15] • Average throughput [39] • Frames per second (FPS) [15] • Failures [25]
User satisfaction metrics		<ul style="list-style-type: none"> • MOS [36] • Number of views [15], • Total play time [15], • Session duration ratio [43], • Abandonment [25], • Engagement [25], • Repeat viewers [25]
(alternately, viewer behavior metrics [25])	<ul style="list-style-type: none"> • Mean opinion score (MOS) [23] 	
Related standards	<ul style="list-style-type: none"> • For cable TV (2004) [20] • For standard television (2004) [18] • For multimedia applications (2008) [22] • Relative to reduced bandwidth reference (2008) [21] • Television (2002) [19] • Multimedia (2008) [23] 	<ul style="list-style-type: none"> • DASH [24] • 3GP-DASH [2]

QoS factors, there could be other factors that wield influence on the QoE of the viewers. For example, considering the vast amount of video content to choose from, the viewers' QoE can be influenced by the type of content being accessed. Further, for a fixed video content, QoE varies significantly by demography, based on age, gender, ethnic background, and language. In addition, seasonal factors, such as the time of day, day of week and season of year, also might influence the QoE of the user towards a particular video content. Finally, there could be many other exogenous factors, such as important local/national/world events, that might contribute to the QoE of a particular viewer.

In the next Section, we describe our proposed cognitive video streaming strategy [40], which considers all the above factors in devising a video streaming strategy. It must be noted that there are no direct comparisons, because the proposed cognitive video streaming architecture is new and the proposed idea of using predicted play time as a surrogate of QoE is also new. However, the three prediction approaches (based on neural networks, survival models and k-nearest neighbor regression) that we discuss in Section IV have some comparisons. For example, [6] uses naive Bayes decision tree and regression methods to predict user engagement from quality metrics and in [12] survival models were used for remaining time prediction.

II. COGNITIVE VIDEO STREAMING

A block diagram of the proposed cognitive video streaming approach is shown in Figure 3. It is com-

prised of three fundamental modules: an *estimation module*, a *prediction module* and an *adaptation module*. The framework is designed in such a way that each module is able to function with some basic functionalities (sub-modules); as more sub-modules are added, the effectiveness of the module and the integrated system is expected to improve. Next, we describe each module in the proposed solution framework.

A. Prediction Module

The nature of completion of a particular video changes from viewer to viewer; some videos are abandoned in the process of “browsing”; some videos are terminated by the viewer because of lengthy buffering and other QoS issues; and some videos are “temporarily” abandoned to be resumed later. Once a viewer starts playing a video, the *remaining play time* of that video is a useful piece of information to the content provider in order to ensure adequate QoE to the viewer. For example, the knowledge of the remaining play time can be used to allocate server bandwidth to the user; it can be used to devise a more appropriate adaptive bitrate switching scheme; and the prior knowledge that a video is possibly terminated by the viewer can be used to recommend more appropriate videos in the first place. At the network level, the predicted play time of each viewing session is useful for managing network traffic.

In addition to QoS, there are several other factors determining the play time ratio (PTR) which is the ratio of the completed time to the actual length of the video ($\text{PTR} \in [0,1]$ is useful to compare the played times of two videos of different length.) However, it

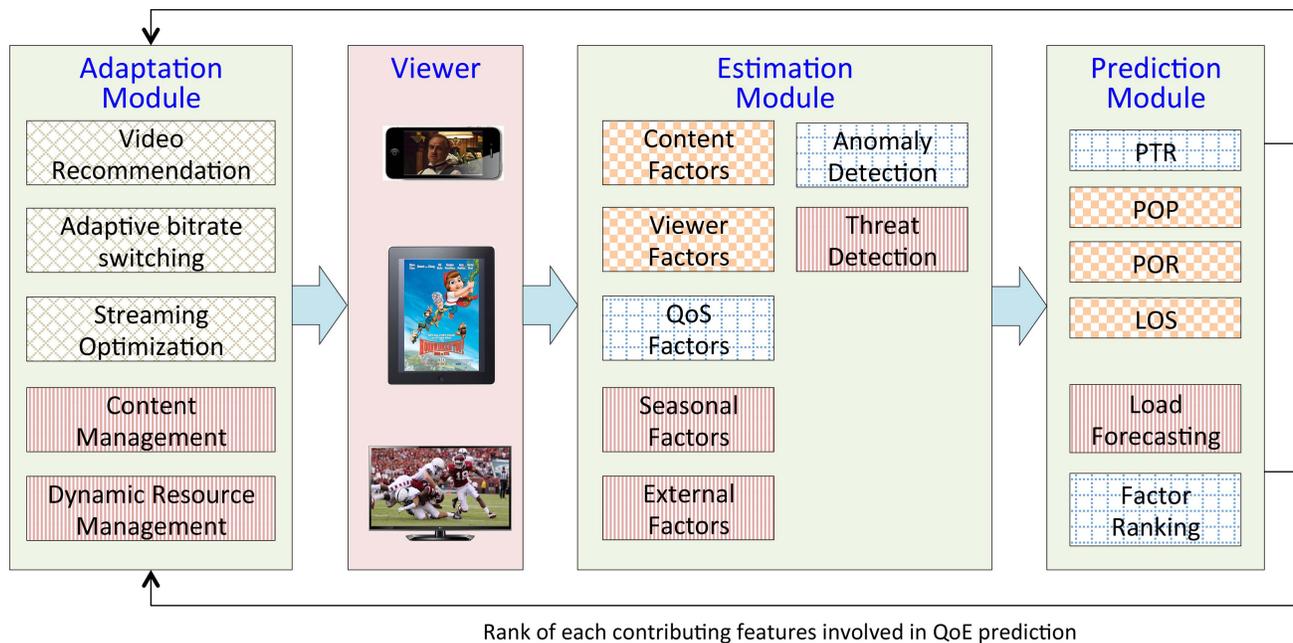


Fig. 3. Proposed Cognitive Video Streaming Architecture.

was reported that shorter videos tend to have higher PTR compared to longer videos [26]; hence, PTR gives better comparison for videos of comparable length. QoS factors such as buffering negatively affect the PTR in well-connected devices. All the relevant factors must be included in order to accurately predict the play time of a video. We divide the factors affecting the PTR into five categories: content-related, viewer-related, QoS-related, seasonal and external. Each factor contains several features affecting the play time; in Table II, we have provided some examples.

Considering all the relevant factors/features helps in accurately predicting the PTR of a particular video session. This also allows us to investigate the features that are significant to PTR prediction. It must be noted that the dominant factor affecting play time will be different from one viewer to the next. Identifying these factors (even after knowing that a particular video has been terminated) will help in devising individualized remedies.

Similar to PTR, there are other user engagement metrics that are indicative of the QoE of a viewer:

- **Probability of return (POR)** tells if the viewer will return to a previously abandoned video. Returning to the same video indicates the importance of that video to the viewer. Hence, POR combined with PTR forms a stronger indicator of the QoE.
- **Probability of re-play (POP)** tells if the viewer will re-play a previously completed video. The difference between POR and POP is that the former is the (probability of) return to an abandoned video and the latter is the (probability of) return to a previously watched video.

TABLE II
Factors Affecting Play-Time Prediction and Sample Features in Each Factor

Factor	Features
Content	popularity, age, length, match to viewer’s preference
Viewer	age, gender, ethnic background, language
QoS	startup time, buffering, average bitrate, throughput
Seasonal	time of day, day of week, season of year
External	important local/national/world events

- **Average length of scrubbing (LOS)** tells how long a particular video will be “scrubbed,” i.e., rewind or forwarded. Scrubbing is the process of moving the player to a different point in the video. For example, most of the viewers might try to scrub past a commercial segment (due to this reason, many video players nowadays disable the scrubbing option during commercial breaks). Apart from commercial breaks, abnormal scrubbing behavior might strongly correlate to the QoE, hence LOS is another effective indicator of QoE.

Later in the paper, we are demonstrating only the PTR prediction. The same algorithms can be used for other three metrics, however, POR, POP and LOS are not computed due to some features missing in the analyzed data.

Developing the ability to understand and predict all the user engagement metrics will help in developing an adaptive streaming method that is responsive to the QoE of the individual viewer (instead of just the QoS factor of a viewer’s device). Another important system variable is load; indeed, **load forecasting** algorithms will

be useful in dynamic resource allocation. In [41], we experimented with Neural networks [30], [38], Nearest neighbor classifiers [35], and Survival modeling [13] techniques in developing a PTR prediction tool. The remainder of this paper is dedicated to PTR prediction. This will be useful in developing the proposed system and the concomitant user-centered QoE prediction models.

B. Estimation Module

The objective of the estimation module is to infer and provide all the features required by the predictive module. First, the estimation module performs the following to prepare the data for training.

- **Anomaly detection:** It is desired to avoid using data containing anomalous events for training. Anomaly detection [8] is also important for accurate feature extraction, security threat detection and QoE monitoring.
- **Threat detection:** Threats are unauthorized usage of content such as accessing unauthorized videos (by sharing login credentials or through other means). Threats are more difficult to detect than anomalies because what constitutes a threat depends on the circumstance. In the VoD domain, threat is an unauthorized usage of content by the subscribers and non-subscribers getting access to content that are not intended to be accesses. Such unauthorized usage is not conducive to the sustained operation of the content provider. The most effective threat detection combines informative features from both anomaly-based and signature-based approaches; understanding of normal (and possibly abnormal) signatures is crucial to devising an effective threat detection strategy.

C. Adaptation Module

The adaptation model consists of the following important sub-modules:

- **Video recommendation:** Video recommendation is an indirect way of improving the QoE of a viewer. Significant attention has been given in the past decade in developing recommendation algorithms. Our proposed methodology will benefit from such recommendation algorithms.
- **Adaptive bitrate switching:** Adaptive bitrate switching strategy helps in achieving uninterrupted play of the video regardless of fluctuating bandwidth (mostly on the user's side).
- **Streaming optimization:** Streaming optimization aims to achieve the most economic usage of bandwidth.
- **Content management:** Content management is required to respond to uneven and unexpected demand of particular video content at particular times.
- **Dynamic resource management:** Dynamic resource allocation [16] helps in optimizing the resources, such as server bandwidth and content, in a way that a

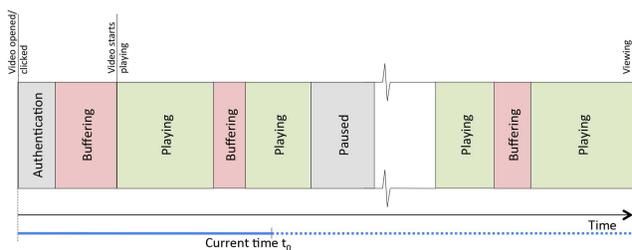


Fig. 4. **Typical video viewing session.** The purpose of the play time prediction tool (PPT) is to estimate the remaining playtime at the current point in time t_0 .

guaranteed QoE can be maintained across all (of the tens of millions of) subscribers.

III. PLAYTIME PREDICTION TOOL (PPT)

In this section, we provide a detailed description of the play time prediction tool [41] of the cognitive video streaming architecture.

Figure 4 shows a typical sequence of events in a viewing session. The session starts when the viewer requests a video. The request may go through an authentication process for non-public videos and then the video starts buffering into the local player. The amount of video being buffered (before the first video frame starts playing) depends on factors, such as the player or the bandwidth. Once a certain portion of the video buffer is filled, the video starts playing in the local player. If the streaming rate is poor, the video player might be forced to temporarily stop playing the video due to an empty buffer. As soon as the buffer is filled again, playing resumes. Nowadays, most streaming protocols use adaptive bitrate switching—meaning the bitrate is adapted dynamically in order to get the best possible video quality for the current bandwidth. The viewing session ends when the entire video is finished playing or when the viewer actively closes that video.

Functionality of the Playtime Prediction Tool (PPT)

In this work, we aim at developing an online *playtime prediction tool (PPT)* that estimates the remaining playtime in a viewing session, see Figure 4. Technically, the tool may run on either the client side or the server side.

To the best of our knowledge, there is no work yet on an online prediction of the session playtime based on an ongoing session. The most similar work [15] aims at developing methods for predicting the playtime of completed sessions.

The PPT presented in this work is the first step in creating a tool that forecasts the entire set of events in a session.

Data used for PPT

In order to perform playtime prediction, the tool exploits protocol data reported by the video player. Typically, this data contains high-level information about

the video session such as in Figure 4. Content related features, e.g., the popularity of the video, also play an important role. A detailed description of the features used in this work will be given in Section V.

Methods

We demonstrate several supervised machine learning approaches for play time prediction. These approaches use previously logged protocol data for training. The proposed play time predictor can be set up for specific users, particular VoD assets, or a group of users.

Benefits

The PPT is of high value to the content provider. First and foremost, it allows the content provider to react before the session is terminated. For example, the content provider can enact counter measures to increase the service quality or recommend alternate content. Even if the PPT predicts a long playtime, the content provider in general could decrease the quality of service to a minimum acceptable level.

Second, the learned playtime prediction model encodes important information about the viewer behavior (of the entire population or even a specific viewer). For example, it is possible to perform a diagnosis that gives the most relevant features that influence the playtime. Also, a playtime prediction model allows for detecting a change in user behavior, and this potentially is of interest when threat detection is the goal.

Last but not least, playtime is a very strong indicator of the QoE. Intuitively, if the QoE is bad, the playtime will be low, too. And if the playtime is long, the QoE cannot be that bad. Hence, a model for the playtime will always be a significant part of a QoE model. In this sense, content providers are interested in increasing the playtime, i.e., the user engagement.

All told, the PPT had a substantial impact on improving the overall QoE of video streaming.

IV. METHODS FOR PLAY-TIME PREDICTION

In this section, we introduce several approaches for playtime prediction at a single specific time t_0 .¹

A. Linear Regression-based Prediction

A simple prediction model of playtime might be a linear combination of the observed features:

$$y_i = \sum_{n=0}^{N_x} k_n x_{i,n} \quad (1)$$

where $x_{i,n}$ is the n th observed feature corresponding to the i th viewing session, and y_i is the playtime. The parameter $\mathbf{k} = [k_0, k_1, k_2, \dots, k_{N_x}]$ can be estimated

by collecting the observation pairs $\{y_i, \mathbf{x}_i\}$ where $\mathbf{x}_i = [1, x_{i,1}, x_{i,2}, \dots, x_{i,N_x}]^T$ for $i = 1, \dots, M$, i.e.,

$$\hat{\mathbf{k}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (2)$$

where $\mathbf{y} = [y_1, y_2, \dots, y_M]^T$ and $\mathbf{X} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_M^T]^T$.

For a given observed feature $\mathbf{x}_j = [1, x_{j,1}, x_{j,2}, \dots, x_{j,N_x}]^T$, the predicted playtime is given as

$$\hat{y}_j = \mathbf{x}_j^T \hat{\mathbf{k}} \quad (3)$$

The linear prediction is useful as a comparison against other nonlinear approaches described later.

B. K-Nearest Neighbor Method

In the k -nearest neighbor approach, the target and feature pairs $\{\mathbf{y}, \mathbf{X}\}$ are kept as training-data. Given the observed feature \mathbf{x}_j , first, the following distance metric is computed

$$d_{i,j} = \mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) \quad (4)$$

where $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j)$ is a distance measure between the arguments \mathbf{x}_i and \mathbf{x}_j . Let \mathbf{y}^k correspond to the play time of the first k of the smallest distance measures. Now, \hat{y}_j is obtained in two different ways: (i) mean of \mathbf{y}^k , (ii) median of \mathbf{y}^k . The median is robust to anomalies and outliers.

C. Survival Models

Survival modeling has found wide application in a number of areas, including medicine [13] and equipment failure analysis [27]. Survival modeling was employed to derive a QoE metric in [12]. In this section, we briefly describe how survival models can be used for playtime prediction.

Let ξ be the time of termination of a particular video. The probability density function of ξ can be written as

$$P_\xi(t) \triangleq f(t) \quad (5)$$

where $f(t)$ is also known as the *survival density function*.

The cumulative probability distribution function of ξ

$$F(t) = P(\xi \leq t) = \int_0^t f(u) du \quad (6)$$

is the fraction of the videos terminated at time t . The remaining (still playing) portion of videos is given by

$$R(t) = P(\xi > t) = 1 - F(t) \quad (7)$$

where $R(t)$ is also known as the *reliability*.

Given that a video has survived until time t , it is often of interest to know the probability that it will be terminated in the next moment, i.e.,

$$h(t) = f(t | \xi > t) = \frac{f(t)}{R(t)} \quad (8)$$

denotes the instantaneous risk or *hazard rate* of the system. Let us rewrite (8) as

$$h(t) = \frac{f(t)}{1 - F(t)} = \frac{F'(t)}{1 - F(t)} = -\frac{R'(t)}{R(t)} \quad (9)$$

¹Hence, we can omit t_0 in the notation used in the remainder of this paper.

Integrating both sides of (9)

$$-\int_0^t h(u)du = \ln R(t) \quad (10)$$

Hence,

$$R(t) = \exp\{-H(t)\} \quad (11)$$

where $H(t) = \int_0^t h(u)du$ is the *cumulative hazard function*.

Using (7) and (11)

$$1 - F(t) = \exp\{-H(t)\} \\ f(t) = h(t)\exp\{-H(t)\} \quad (12)$$

So far it has been assumed that $f(t)$ (and hence $R(t)$ and $h(t)$) are all functions of time only. However, all of these functions are dependent on features $\mathbf{x} = \{\mathbf{x}_i\}$, or *covariates*. The *proportional hazard function*, proposed by Cox [13], suggests to separate the time-dependent and feature-dependent hazards as follows:

$$h(t, \mathbf{x}) = \lambda(t)\exp\{\mathbf{b}^T \mathbf{x}\} \quad (13)$$

where $\lambda(t)$ is the baseline time-dependent hazard function, x_i is the covariate, and b_i is the coefficient corresponding to the i th covariate, x_i .

Now, (11) and (12) are rewritten as

$$f(t) = \lambda(t)\exp\{\mathbf{b}^T \mathbf{x} - \Lambda(t)e^{\mathbf{b}^T \mathbf{x}}\} \quad (14)$$

$$R(t) = \exp\{-\Lambda(t)e^{\mathbf{b}^T \mathbf{x}}\} \quad (15)$$

where $\Lambda(t) = \int_0^t \lambda(u)du$. Cox suggested that the the model parameters \mathbf{b} can be estimated independent of $\lambda(t)$ by maximizing the partial likelihoods. Once \mathbf{b} is estimated, there are several approaches in the literature to model and estimate (the parameters of) $\lambda(t)$.

Once the parameters are estimated, the remaining play time at time u can be computed as

$$\hat{y}_j(u) = \frac{\int_u^\infty (t-u)f_j(t)dt}{R_j(u)} \quad (16)$$

where $f_j(t)$ and $R_j(u)$ are obtained by substituting \mathbf{x}_j for \mathbf{x} in (14) and (15), respectively, and u is the time elapsed.

An advantage of the survival model-based approaches described above is that the playtime prediction can be updated as the video progresses. In this paper, we assume $\lambda(t) = \lambda$.

D. Neural Networks

The playtime can be modeled as a function of the observed features using artificial neural networks (e.g., multi-layer perceptrons)

$$y_i = f(\mathbf{x}_i, \{w_{l,k}\}_{l=1, k=1}^{N_L, N_h}) \quad (17)$$

where $w_{l,k}$ are different weights and N_L is the number of layers and N_h is the number of hidden nodes. Given a set of (past) training data \mathbf{y}, \mathbf{X} , there are several approaches

to learn the weights [37]. A trained neural network can be used to predict the playtime for a given feature set \mathbf{x}_j .

Neural Network predictor was implemented by the use of the built in neural network function in Matlab™. The number of neurons and the number of hidden layers are selected to be the ones to give the highest prediction accuracy metrics with the training data. For the particular example described in Section V, a multi-layer perceptron model was selected with three hidden layers each having six neurons.

V. SIMULATION STUDIES

In this section, we evaluate the proposed approaches using data from 8808 viewing sessions. In order to avoid any confounding effects, all these 8808 viewing sessions are selected from the same type of video; in particular, all these videos are selected to be the episodes of ‘‘The Simpsons.’’ Further, all these videos were viewed on the same day. We focus on the first 8 minutes as we try to understand early quitters due to the low streaming quality. A portion of these sessions is randomly selected and denoted as the ‘‘learning’’ dataset, and the rest is kept for testing. Each feature in the testing data is used for predicting its playtime. This procedure is repeated for 10 Monte-Carlo runs.

Our work is based on a dataset from the VoD streaming service *Xfinity On Demand* from *Comcast*. The available data was logged by the video players and consists of a sequence of events that come with time stamps, device ids, and further information. Specifically, we use the following logged events from each user. For each of these events, the starting and ending times are available.

- *Opening*: Indicates that a new viewing session is opened by the user.
- *Playing*: Video starts playing.
- *Buffering*: The player starts buffering; the video doesn’t play until a certain amount of data is buffered. Further, the buffering event can occur while a video is playing.
- *Paused*: The pause event occurs when the user presses the pause button.
- *Closing*: Video may stop playing either due to the user ending the session or when the end of the video is reached.
- *Bitrate switched*: This event occurs whenever the streaming bitrate changes.

We define a viewing session as the events between the opening and closing events at a particular device. Based upon the above described events, we determine the following session features that potentially affect the playtime and the QoE.

A. Data Analysis and Visualization

The following features are used in our current analysis:

- 1) Number of buffering events (f_1)

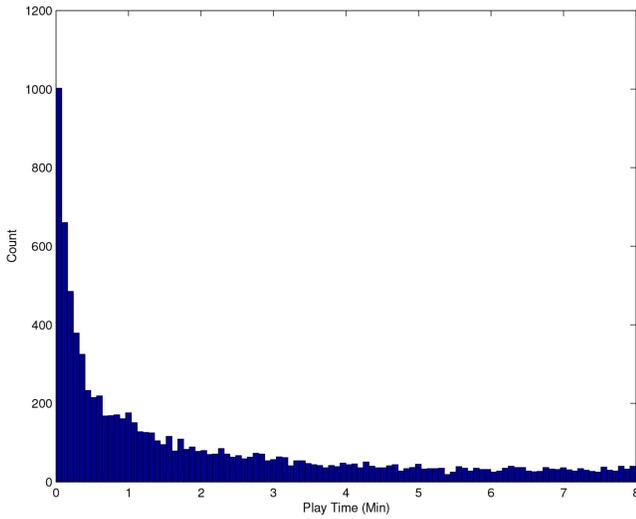
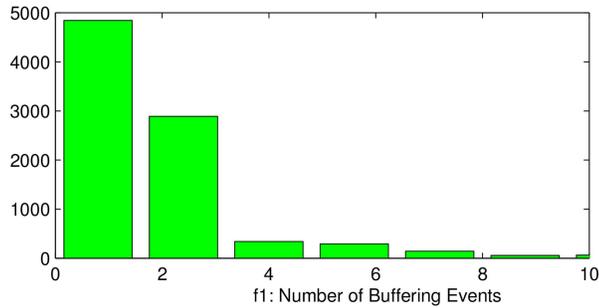


Fig. 5. Histogram of playtime.

- 2) Number of paused events (f_2)
- 3) Inter buffering time (f_3): The average time (in seconds) between two buffering events.



- 4) Startup time (f_4): The time it takes from when the user hits the play button to the time the video starts playing on the screen.
- 5) Average bit rate (f_5): The average bit rate is measured in Mega bits per second (Mbps).
- 6) Buffering ratio (f_6): the relation between the total buffering time and the total play time of a video. The buffering ratio negatively affects the QoE.

Figure 5 shows the histogram of playtime for all the 8808 viewing sessions. The play time distribution suggests an exponential decay in this case. Figure 6 shows the histograms of the corresponding features. It can be seen that the majority of the video sessions had up to two buffering and paused events each. The startup time is approximately 4 seconds for the majority of the videos. The peaks around the 1.8 Mbps and 4.2 Mbps indicate the presence of standard video and high definition video, respectively.

B. Performance Metrics

In this section, we use the algorithms introduced in Section IV for playtime prediction and assess their performance. Due to lack of knowledge on the statistical

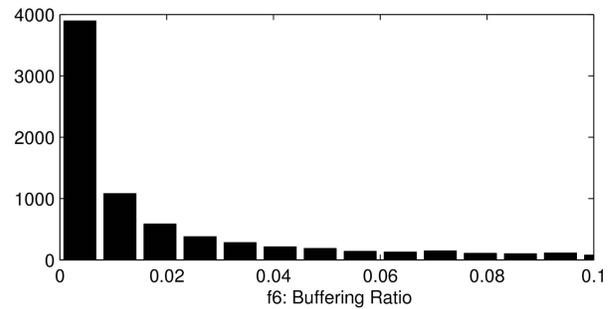
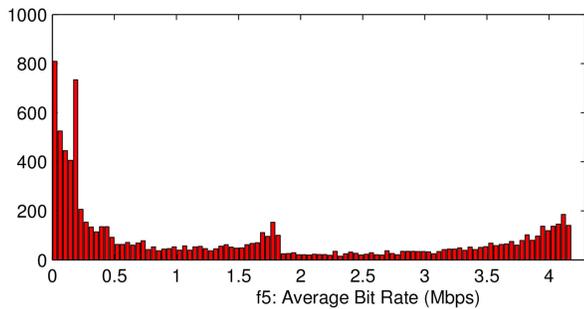
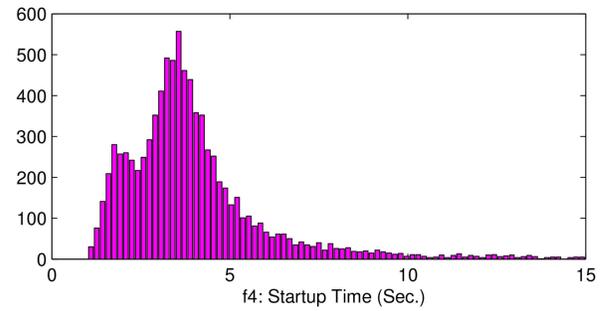
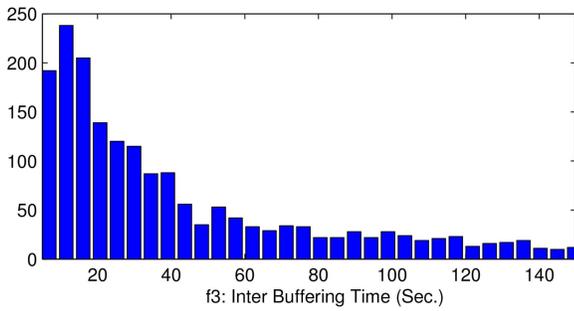
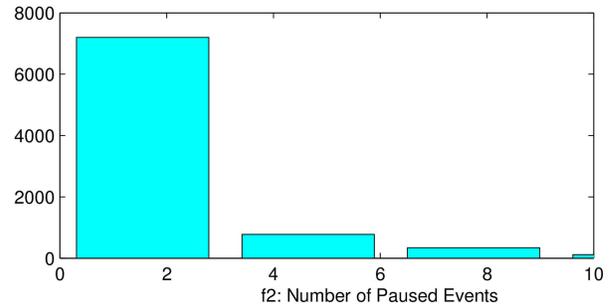


Fig. 6. Histogram of features.

properties of playtime, we suggest using several surrogate metrics for assessing playtime. The following four metrics were considered.

1) *Normalized Mean-Squared Error (NMSE)*: This metric gives insight on the error in playtime prediction and is given by

$$\text{NMSE} = \frac{1}{M} \sum_{i=1}^M \left(\frac{y_i - \hat{y}_i}{y_i} \right)^2 \quad (18)$$

2) *R² Fit*: The coefficient of determination, R^2 , gives insight into how well the data points fit the statistical model used to predict playtime. A value of $R^2 = 1$ indicates perfect fit, and smaller the R^2 , the poorer is the fit.

$$R^2 = 1 - \frac{\sum_{i=1}^M (y_i - \hat{y}_i)^2}{\sum_{i=1}^M (y_i - \bar{y})^2} \quad (19)$$

where $\bar{y} = (1/M) \sum_{i=1}^M y_i$.

3) *Ratio of Predicted and True Playtime Greater than r*: The playtime is a quantity that can generally vary anywhere from less than 1 minute to several hours. A prediction error of 1 min is significant if the actual play time is 5 min; however, it is not so significant if the actual play time is 2 hours. The NMSE captures this through normalization; however, the following metric captures this error in a different light.

$$\text{RG}(r) = \frac{\# \left\{ \frac{\hat{y}_i}{y_i} > r \right\}}{M} \quad (20)$$

where $\#\{\cdot\}$ denotes the number of times the argument is true.

4) *Ratio of Predicted and True Playtime Less than 1/r*: Similar to $\text{RG}(r)$, the following metric captures the instances when the prediction was significantly smaller than the true value of playtime.

$$\text{RL}(1/r) = \frac{\# \left\{ \frac{\hat{y}_i}{y_i} < \frac{1}{r} \right\}}{M} \quad (21)$$

C. Feature Selection

With N features, there are $2^N - 1$ possible subsets of features. Although it might be thought that more is better, in machine learning, one can be subject to the ‘‘curse of dimensionality’’: extra features that are uninformative actually hurt prediction performance by ‘‘fitting to the noise.’’ In Figures 7, 8, 9 and 10, we show the performance(s) plotted against binary representation of feature combinations, from 1 to $2^N - 1$. Each time, half the dataset is randomly selected and used for learning and the playtime is predicted using the rest of the data. This procedure is repeated for 10 Monte-Carlo runs (This is called a 10×2 cross validation.) and the median of each of the metrics is plotted in Figures 7–10. There are six subplots in each of Figures 7–10, showing the results of different playtime prediction approaches: Survival modeling, k-nearest neighbor (mean), k-nearest

TABLE III
Performance Metrics

Feature Combination ID	f1	f2	f3	f5	f6	f7	R2	NMSE	RG(2)	RL(0.5)
61	x	x	x	x	x	x	0.74479	1.1028	0.19743	0.074251
55	x	x	x	x	x	x	0.73473	1.1427	0.22604	0.073569
63	x	x	x	x	x	x	0.71984	1.1738	0.24353	0.068233
57	x	x	x	x	x	x	0.71637	1.1788	0.25386	0.059378
59	x	x	x	x	x	x	0.7162	1.1766	0.25079	0.065054
49	x	x	x	x	x	x	0.71531	1.1683	0.2584	0.066417
61	x	x	x	x	x	x	0.74479	1.1028	0.19743	0.074251
55	x	x	x	x	x	x	0.73473	1.1427	0.22604	0.073569
49	x	x	x	x	x	x	0.71531	1.1683	0.2584	0.066417
63	x	x	x	x	x	x	0.71984	1.1738	0.24353	0.068233
59	x	x	x	x	x	x	0.7162	1.1766	0.25079	0.065054
57	x	x	x	x	x	x	0.71637	1.1788	0.25386	0.059378
61	x	x	x	x	x	x	0.74479	1.1028	0.19743	0.074251
37	x	x	x	x	x	x	0.69295	1.2289	0.2072	0.070391
36	x	x	x	x	x	x	0.54402	1.4849	0.21946	0.067666
55	x	x	x	x	x	x	0.73473	1.1427	0.22604	0.073569
45	x	x	x	x	x	x	0.65841	1.2883	0.23399	0.073683
63	x	x	x	x	x	x	0.71984	1.1738	0.24353	0.068233
41	x	x	x	x	x	x	0.65943	1.2865	0.28213	0.043824
53	x	x	x	x	x	x	0.71386	1.1796	0.271	0.057788
57	x	x	x	x	x	x	0.71637	1.1788	0.25386	0.059378
39	x	x	x	x	x	x	0.67346	1.2644	0.25318	0.060513
59	x	x	x	x	x	x	0.7162	1.1766	0.25079	0.065054
33	x	x	x	x	x	x	0.65471	1.2997	0.31903	0.065622
Total Features	23	10	14	14	18	24				

TABLE IV
Feature Ranking Based on Borda Count

Feature ID	Rank: R2	Rank: NMSE	Rank: RG(2)	Rank: RL(0.5)	Borda Count	Borda Rank
61	1	1	1	13	40	1
55	2	2	4	11	37	2
57	4	6	9	3	34	3
59	5	5	7	5	34	3
63	3	4	6	9	34	3
49	6	3	10	7	30	4
53	7	7	11	2	29	5
37	8	8	2	10	28	6
39	9	9	8	4	26	7
41	10	10	12	1	23	8
36	13	13	3	8	19	9
45	11	11	5	12	17	10
33	12	12	13	6	13	11

neighbor (median), LS, neural networks and random. In ‘‘random’’ approach, we randomly select a playtime from the training dataset.

Next we select just one playtime prediction approach shown in Figures 7–10 and try to select the best feature set (out of $2^N - 1$) for online prediction. We select the neural networks approach for this evaluation. The objective of feature selection is to find the features that gives the best result across all performance metrics defined in Section V-B.

Table III shows the first six feature sets ranked according to each of the performance metrics: R^2 , NMSE, RG(2) and RL(0.5). For example, the features corresponding to the binary number 61, i.e., NRB, IBT, STT, BR and BUR, give the best performance according to R^2 , NMSE and RG(2), whereas the features corresponding to the binary number 41, i.e., NRB, STT, and BUR, give the best performance according to RL(0.5).

We employ a method known as *Borda count* [9] in order to select the best feature subset based on all four evaluation metrics. For each feature ID (binary number) in Table III, the Borda count gives a point based on the ranking of that ID using each of the four evaluation metrics. Then, the feature ID having the most Borda points is selected as the best feature set in terms of all four evaluation metrics. Table IV summarizes the Borda count procedure in selecting the best feature subset. For this particular example, the feature subset with ID 61 is ranked first, while the one with ID 41 is ranked 8th.

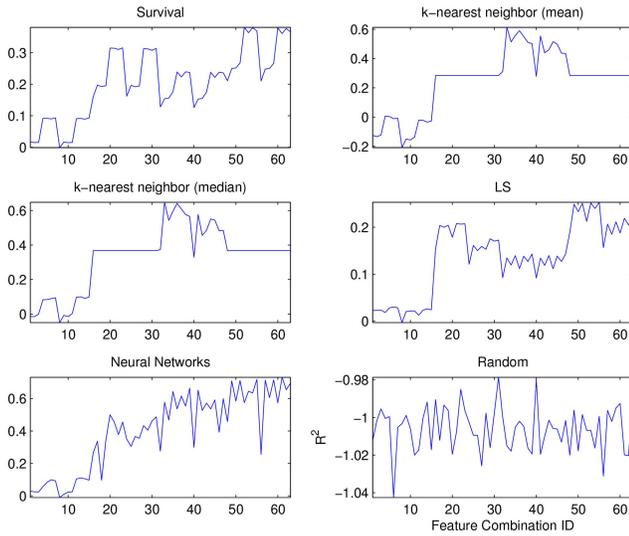


Fig. 7. Median R^2 Fit.

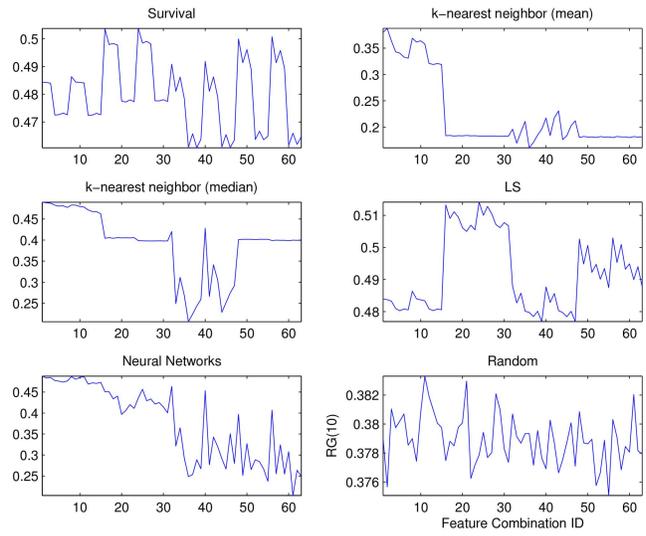


Fig. 9. Median $RG(r)$.

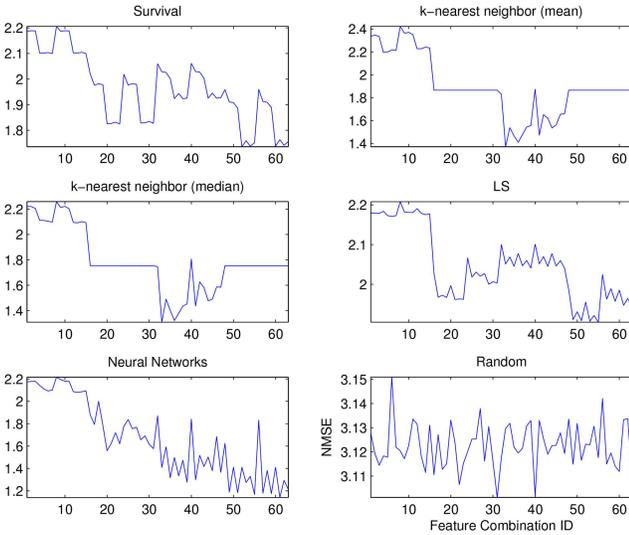


Fig. 8. Median NMSE.

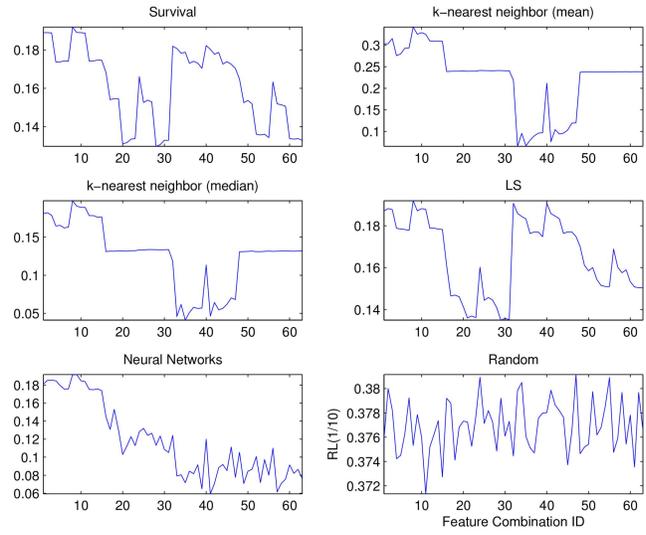


Fig. 10. Median $RL(1/r)$.

Hence, for neural network approach, the features NRB, IBT, STT, BR and BUR will be used for online playtime prediction. The features are selected in a similar fashion for the rest of the five playtime prediction methods.

D. Playtime Prediction Results

Assuming that the best features are selected offline based on the approach described in the previous section, in this section we show the online playtime prediction results of each approach.

Figure 11 shows a scatter plot of true vs. predicted play time. Each subplot corresponds to the prediction approach mentioned in the title. For each approach, the feature ID corresponding to the top Borda count is displayed in parenthesis as well. Ideally, the scatter plot should look like a line from the origin with gradient 1; the “thickness” of the scatter as well as “concentrations” at off-diagonal places indicate the error in predictions.

Figure 12 shows the predicted play time as an overlay plot of true and estimates; the blue line shows the actual play time and the red stars are the predicted ones.

Figure 13 shows the prediction errors as a histogram; a “thin” histogram implies good prediction and vice versa. Out of all the six approaches, the neural network approach yields the best prediction results followed by both of the k-nearest neighbor methods. The “random” approach is shown as a measure of comparison to the worst method; the random approach randomly picks a data from the training set as the predicted play time.

The playtime prediction results shown in this section demonstrate that the proposed approaches are promising, since they all perform better than the random prediction approach. However, our objective in this paper is not to develop a perfect play time prediction tool, rather to demonstrate a functioning PPT, which is a component of the proposed cognitive video streaming architecture.

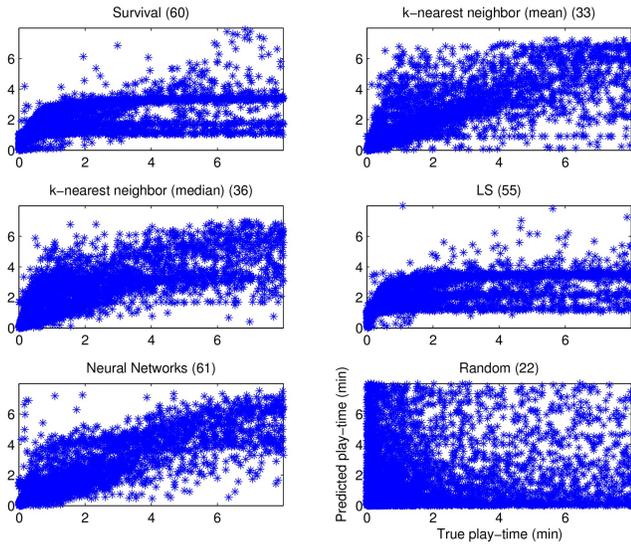


Fig. 11. Scatter plot of true vs. predicted playtime.

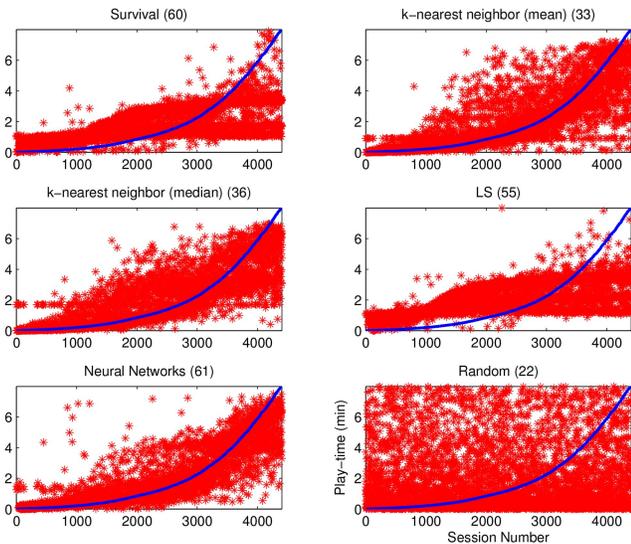


Fig. 12. Overlay plot of true vs. predicted playtime.

For more accurate playtime prediction results, the PPT has to be “fed” with more than the QoS related features, such as the content related features, the viewer related features and the external features. In this paper, we limited the discussion to outlining the technical details and demonstration of the PPT with just QoS related features.

VI. CONCLUSIONS AND DISCUSSION

The contribution of the paper is to describe a framework for modeling the variables that may affect the quality of experience (QoE) of video-on-demand (VoD) services, with the aim of maximizing QoE for subscribers of commercial media providers. However, QoE is difficult to measure based on the usage data that is available to the service provider: it is only indirectly inferable. Hence, one contribution of this paper is to

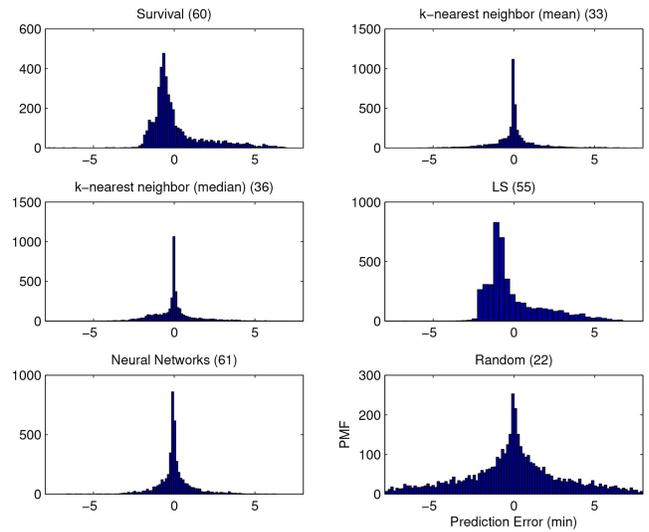


Fig. 13. PMF of playtime-prediction error.

formulate the problem and discuss the relevant literature, much of which appears in quite diverse journals. Implicit is that if QoE is well understood, its real-time prediction might be used to adapt the underlying content delivery strategy via adaptive bit-rate switching, streaming optimization, content management, dynamic resource management and video recommendations.

We proposed and discussed various QoE measures, such as *playtime*, *probability of return*, *probability of replay* and *average length of scrubbing*. We discussed approaches for the prediction of such QoE measures by supervised machine learning algorithms. Further, we described the type of features that can be computed from the subscriber data for use in the QoE prediction algorithms as predictive features. We categorized these features into *content-related features*, *viewer-related features* and *quality of service (QoS)-related features*. Then, we demonstrated playtime prediction through several supervised classification approaches using QoS related features that are collected from the subscribers of a popular VoD service.

The proposed cognitive video streaming architecture is suitable to future developments in the fast changing video consumption arena. For example, more accurate measure of the QoE can be obtained by making use of other relevant data. In [31], [32] and [34], we used the eye tracking data, such as pupil dilation and eye-gaze pattern in order to estimate the cognitive context of unmanned aerial system (UAS) operators, while they execute reconnaissance missions. However, existing VoD systems are not equipped to measure/collect eye tracking data. Considering the fact that most of the video playing devices (with the exception of TV) are equipped with a front facing camera, creating and exploiting eye tracking data for QoE estimation has a good chance of becoming a reality.

The availability of additional physiological measurements, such as heart rate, breathing rate and body temperature will pave the way for improved accuracy in QoE estimation. None of the existing video devices are equipped with the sensors to measure these physiological features. However, driven by the personal health monitoring devices (also known as fitness trackers, such as fitbit™), the (direct or indirect) availability of these physiological measurements for VoD services could become a reality in the future. If and when that happens, the QoE estimation can be done with increased confidence and the cognitive video streaming architecture will be able to cater to more advanced form of entertainment.

In general, QoE estimation is a part of the much larger human machine systems (HMS). An HMS is formed when a semi-autonomous system is operated by a human (or group of humans) operator(s); (i) a pilot flying an aircraft, (ii) a car driven by a human driver, (iii) a person working on a computer, and (iv) an unmanned aerial system (UAS) mission executed by a group of operators are all examples of HMS. It has been well understood that human performance becomes suboptimal when the workload is too high as well as when it is too low [46]. An important research challenge in the HMS domain is to create machines that are able to better understand human behavior so that the overall efficiency of the HMS can be improved through increased productivity and reduced safety risk [33]. An understanding of the physiological behavior of the human body can be combined with statistical machine learning theory in order to develop algorithms that are able to accurately predict the cognitive context (such as the difficulty of work, level of alertness, etc) experienced by humans. Also, a more generalized topic of context based information fusion [47] gives additional insights on this emerging research field.

The future of VoD system will look more similar to a HMS with one exception: the objective of all the other HMS is to perform a certain task with high efficiency, whereas the objective of the VoD system is to offer entertainment and pleasure to the human. A new ecosystem of services and applications are waiting to be developed around a successful VoD system. For example, doctors might prescribe certain videos as part of treatment plans; students might be asked to watch a certain video as part of a curriculum, all under the assumption that a dependable QoE estimation system (in general terms, a cognitive context detection system) is available. The immediate challenge of the information fusion community is to develop such systems.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for constructive reviews of this work. This research was funded by Comcast Corporation and the

U.S. Office of Naval Research (ONR) under contracts #PO 998604 and #N00014-16-1-2036. Peter Willett was supported by the Naval Postgraduate School via ONR contract #N00244-16-1-0017.

REFERENCES

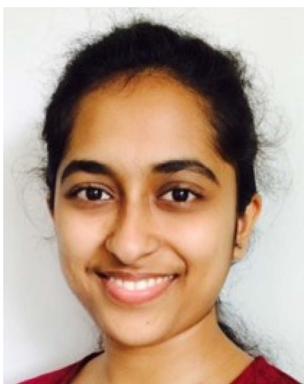
- [1] “Cisco visual networking index: Forforecast methodology 2013–2018.” Cisco, Tech. Rep., June 2014.
- [2] *Transparent End-to-End Packet Switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming Over HTTP (3GP-DASH)*, 3GPP Std. TS 26.247 v10.1.0, June 2011.
- [3] V. Aggarwal, E. Halepovic, J. Pang, S. Venkataraman, and H. Yan
Prometheus: Toward quality-of-experience estimation for mobile apps from passive network measurements, *ACM HotMobile*, 2014.
- [4] A. Anand, A. Balachandran, A. Akella, V. Sekar, and S. Seshan
Enhancing video accessibility and availability using information-bound references, in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*. ACM, 2013, pp. 345–356.
- [5] A. Balachandran, V. Sekar, A. Akella, and S. Seshan
Analyzing the potential benefits of CDN augmentation strategies for internet video workloads, in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 2013, pp. 43–56.
- [6] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang
A quest for an internet video quality-of-experience metric, in *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*. ACM, 2012, pp. 97–102.
- [7] ———
Developing a predictive model of quality of experience for internet video, in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 339–350.
- [8] B. Balasingam, S. Sankavaram, K. Choi, D. M. Ayala, D. Sidoti, K. R. Pattipati, P. Willett, C. Lintz, G. Commeau, F. Dorigo, and J. Fahrny
Online anomaly detection in big data, in *International Conference on Information Fusion*, 2014.
- [9] D. Black
Partial justification of the Borda count, *Public Choice*, vol. 28, no. 1, pp. 1–15, 1976.
- [10] P. Brooks and B. Hestnes
User measures of quality of experience: why being objective and quantitative is important, *IEEE Network*, vol. 24, no. 2, pp. 8–13, March 2010.
- [11] J. Chen, A. Ghosh, J. Magutt, and M. Chiang
QAVA: quota aware video adaptation, in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. ACM, 2012, pp. 121–132.
- [12] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei
Quantifying Skype user satisfaction, in *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4. ACM, 2006, pp. 399–410.
- [13] D. R. Cox and D. Oakes
Analysis of survival data. CRC Press, 1984, vol. 21.

- [14] N. Damera-Venkata, T. D. Kite, W. S. Geisler, B. L. Evans, and A. C. Bovik
Image quality assessment based on a degradation model, *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 636–650, 2000.
- [15] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. A. Joseph, A. Ganjam, J. Zhan, and H. Zhang
Understanding the impact of video quality on user engagement, *SIGCOMM-Computer Communication Review*, vol. 41, no. 4, p. 362, 2011.
- [16] X. Han, S. Mandal, K. Pattipati, D. Kleinman, and M. Mishra
An optimization-based distributed planning algorithm: A blackboard-based collaborative framework, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 6, pp. 673–686, June 2014.
- [17] T.-Y. Huang, R. Johari, and N. McKeown
Downton Abbey without the hiccups: Buffer-based rate adaptation for http video streaming, in *Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking*. ACM, 2013, pp. 9–14.
- [18] *Objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference*, International Telecommunication Union Std., ITU-R Recommendation BT.1683, 2004.
- [19] *Methodology for the subjective assessment of the quality of television pictures*, International Telecommunication Union Std., ITU-R Recommendation BT.500-11, 2002.
- [20] *Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference*, International Telecommunication Union Std., ITU-T Recommendation J.144, 2004.
- [21] *Perceptual visual quality measurement techniques for multimedia services over digital cable television networks in the presence of a reduced bandwidth reference*, International Telecommunication Union Std., ITU-T Recommendation J.246, 2008.
- [22] *Objective perceptual multimedia video quality measurement in the presence of a full reference*, International Telecommunication Union Std., ITU-T Recommendation J.247, 2008.
- [23] *Subjective video quality assessment methods for multimedia applications*, International Telecommunication Union Std., ITU-T Recommendation P.910, 2008.
- [24] *Information Technology—Dynamic Adaptive Streaming Over HTTP (DASH)—Part 1: Media Presentation Description and Segment Formats*, ISO Std. IEC DIS 23 009-1, Aug 2011.
- [25] S. S. Krishnan and R. K. Sitaraman
Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs, in *Proceedings of the 2012 ACM conference on Internet measurement conference*. ACM, 2012, pp. 211–224.
- [26] S. Krishnan and R. Sitaraman
Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs, *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 2001–2014, Dec 2013.
- [27] W. C. Levy, D. Mozaffarian, D. T. Linker, S. C. Sutradhar, S. D. Anker, A. B. Cropp, I. Anand, A. Maggioni, P. Burton, M. D. Sullivan et al.
The Seattle heart failure model prediction of survival in heart failure, *Circulation*, vol. 113, no. 11, pp. 1424–1433, 2006.
- [28] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. Begen, and D. Oran
Probe and adapt: Rate adaptation for HTTP video streaming at scale, *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 719–733, April 2014.
- [29] C. Liu, R. K. Sitaraman, and D. Towsley
“Go-with-the-winner: Client-side server selection for content delivery,” *arXiv preprint arXiv:1401.0209*, 2013.
- [30] R. Lynch and P. Willett
Bayesian classification and feature reduction using uniform Dirichlet priors, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 33, no. 3, pp. 448–464, June 2003.
- [31] P. Mannaru, B. Balasingam, K. Pattipati, C. Sibley, and J. Coyne
Cognitive context detection in UAS operators using gaze patterns, in *SPIE Conferences on Defense, Security, and Sensing*, April 2016.
- [32] ———
Cognitive context detection in UAS operators using pupillary measurements, in *SPIE Conferences on Defense, Security, and Sensing*, April 2016.
- [33] ———
Human-machine system improvement through cognitive context detection, in *Annual Meeting of the Human Factors and Ergonomics Society*, Sept. 2016.
- [34] ———
On the use of hidden Markov models for eye-gaze pattern modeling and classification, in *SPIE Conferences on Defense, Security, and Sensing*, April 2016.
- [35] S. Marano, V. Matta, and P. Willett
Nearest-neighbor distributed learning by ordered transmissions, *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5217–5230, Nov 2013.
- [36] R. K. Mok, E. W. Chan, and R. K. Chang
Measuring the quality of experience of HTTP video streaming, in *IFIP/IEEE International Symposium on Integrated Network Management*. IEEE, 2011, pp. 485–492.
- [37] K. P. Murphy
Machine learning: a probabilistic perspective. MIT press, 2012.
- [38] C. Neukirchen, J. Rottland, D. Willett, and G. Rigoll
A continuous density interpretation of discrete HMM systems and MMI-neural networks, *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 4, pp. 367–377, May 2001.
- [39] O. Oyman and S. Singh
Quality of experience for HTTP adaptive streaming services, *IEEE Communications Magazine*, vol. 50, no. 4, pp. 20–27, 2012.
- [40] D. Pasupuleti, P. Mannaru, B. Balasingam, M. Baum, K. R. Pattipati, and P. Willett
Cognitive video streaming, in *International Conference on Electrical, Electronics, Engineering Trends, Communication, Optimization and Sciences*, March 2015.

- [41] D. Pasupuleti, P. Mannaru, B. Balasingam, M. Baum, K. R. Pattipati, P. Willett, C. Lintz, G. Commeau, F. Dorigo, and J. Fahrny
Online playtime prediction for cognitive video streaming, in *IEEE International Conference on Information Fusion*, July 2015.
- [42] M. H. Pinson and S. Wolf
A new standardized method for objectively measuring video quality, *IEEE Transactions on Broadcasting*, vol. 50, no. 3, pp. 312–322, 2004.
- [43] F. Qiu and Y. Cui
A quantitative study of user satisfaction in online video streaming, in *Consumer Communications and Networking Conference*. IEEE, 2011, pp. 410–414.
- [44] J. Shaikh, M. Fiedler, T. Minhas, P. Arlos, and D. Collange
Passive methods for the assessment of user-perceived quality of delivery, in *SNCNW*, 2011, p. 73.
- [45] S.-H. Shen and A. Akella
An information-aware QoE-centric mobile video cache, in *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013, pp. 401–412.
- [46] C. Sibley, J. Coyne, and J. Morrison
Research considerations for managing future unmanned systems. 2015, in *AAAI Spring Symposium on Foundations of Autonomy and Its (Cyber) Threats: From Individuals to Interdependence*. AAAI Press, 2015.
- [47] L. Snidaro, J. García, and J. Llinas
Context-based information fusion: a survey and discussion, *Information Fusion*, vol. 25, pp. 16–31, 2015.
- [48] C. J. Van den Branden Lambrecht and O. Verscheure
Perceptual quality measure using a spatiotemporal model of the human visual system, in *Electronic Imaging: Science & Technology*. International Society for Optics and Photonics, 1996, pp. 450–461.
- [49] M. Venkataraman and M. Chatterjee
Quantifying video-QoE degradations of internet links, *IEEE/ACM Transactions on Networking*, vol. 20, no. 2, pp. 396–407, April 2012.
- [50] Z. Wang and A. C. Bovik
Mean squared error: love it or leave it? a new look at signal fidelity measures, *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009.
- [51] Z. Wang, L. Lu, and A. C. Bovik
Video quality assessment based on structural distortion measurement, *Signal processing: Image communication*, vol. 19, no. 2, pp. 121–132, 2004.



Devaki Pasupuleti received her M.S. in Electrical Engineering from University of Connecticut, Storrs in 2015. Since 2015, she has been a Software Engineer at Cisco Systems, San Jose. Her research interests are in Big Data, Machine learning, Networking Protocols, Software Defined Networking, distributed information fusion and their applications in automated systems. Devaki Pasupuleti has authored 2 conference papers in these areas.



Pujitha Mannaru received the B.E. degree in Electronics and Communications Engineering from PES Institute of Technology—Bangalore South Campus, affiliated to Visvesvaraya Technological University, India, in 2013. She is currently pursuing her Ph.D. in Systems Engineering at the University of Connecticut, Storrs, USA. Her research interests are in the areas of applications of signal processing, machine learning, and psychophysiological measures to proactive decision support tools and human-machine systems.



Balakumar Balasingam received his M.A.Sc. and Ph.D. in Electrical Engineering from McMaster University, Canada in 2004 and 2008, respectively. He held a postdoctoral position at the University of Ottawa from 2008 to 2010, and then a University Postdoctoral position at the University of Connecticut from 2010 to 2012. Since 2012, he has been an Assistant Research Professor in the Department of Electrical and Computer Engineering at the University of Connecticut. His research interests are in signal processing, machine learning, and distributed information fusion and their applications in cyber-physical systems, cyber-human systems and human-machine systems. Dr. Balasingam has authored over 50 research publications in these areas.



Marcus Baum is a Juniorprofessor (Assistant Professor) at the University of Goettingen, Germany. He received the Diploma degree in computer science from the University of Karlsruhe (TH), Germany, in 2007, and graduated as Dr.-Ing. (Doctor of Engineering) at the Karlsruhe Institute of Technology (KIT), Germany, in 2013. From 2013 to 2014, he was postdoc and assistant research professor at the University of Connecticut, CT, USA. His research interests are in the field of data fusion, estimation, and tracking. Marcus Baum is associate administrative editor of the *Journal of Advances in Information Fusion* (JAIF) and serves as local arrangement chair of the 19th International Conference on Information Fusion (FUSION 2016). He received the best student paper award at the FUSION 2011 conference.

Krishna Pattipati received the B. Tech. degree in electrical engineering with highest honors from the Indian Institute of Technology, Kharagpur, in 1975, and the M.S. and Ph.D. degrees in systems engineering from UConn, Storrs, in 1977 and 1980, respectively. He was with ALPHATECH, Inc., Burlington, MA from 1980 to 1986. He has been with the department of Electrical and Computer Engineering at UConn, where he is currently the Board of Trustees Distinguished Professor and the UTC Chair Professor in Systems Engineering. Dr. Pattipati's research activities are in the areas of proactive decision support, uncertainty quantification, smart manufacturing, autonomy, knowledge representation, and optimization-based learning and inference. A common theme among these applications is that they are characterized by a great deal of uncertainty, complexity, and computational intractability. He is a cofounder of Qualtech Systems, Inc., a firm specializing in advanced integrated diagnostics software tools (TEAMS, TEAMS-RT, TEAMS-RDS, TEAMATE), and serves on the board of Aptima, Inc.

Dr. Pattipati was selected by the IEEE Systems, Man, and Cybernetics (SMC) Society as the Outstanding Young Engineer of 1984, and received the Centennial Key to the Future award. He has served as the Editor-in-Chief of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B from 1998 to 2001, Vice-President for Technical Activities of the IEEE SMC Society from 1998 to 1999, and as Vice-President for Conferences and Meetings of the IEEE SMC Society from 2000 to 2001. He was co-recipient of the Andrew P. Sage Award for the Best SMC Transactions Paper for 1999, the Barry Carlton Award for the Best AES Transactions Paper for 2000, the 2002 and 2008 NASA Space Act Awards for "A Comprehensive Toolset for Model-based Health Monitoring and Diagnosis," and "Real-time Update of Fault-Test Dependencies of Dynamic Systems: A Comprehensive Toolset for Model-Based Health Monitoring and Diagnostics," and the 2003 AAUP Research Excellence Award at UCONN. He also won the best technical paper awards at the 1985, 1990, 1994, 2002, 2004, 2005 and 2011 IEEE AUTOTEST Conferences, and at the 1997, 2004 Command and Control Conference. He is an elected Fellow of IEEE and of the Connecticut Academy of Science and Engineering.





Peter Willett (F'03) received his B.A.Sc. (engineering science) from the University of Toronto in 1982, and his Ph.D. degree from Princeton University in 1986. He has been a faculty member at the University of Connecticut ever since, and since 1998 has been a Professor. His primary areas of research have been statistical signal processing, detection, machine learning, data fusion and tracking. He also has interests in and has published in the areas of change/abnormality detection, optical pattern recognition, communications and industrial/security condition monitoring. He is editor-in-chief of IEEE Signal Processing Letters. He was editor-in-chief for IEEE Transactions on Aerospace and Electronic Systems (2006–2011), and was Vice President for Publications for AESS (2012–2014). He is a member of the IEEE AESS Board of Governors 2003–2009, 2011–2016. He was General Co-Chair for the 2006 IEEE/ISIF Fusion Conference in Florence, Italy, and again for both 2008 in Cologne, Germany and 2011 in Chicago IL. He was Program Co-Chair for the 2016 ISIF/IEEE Fusion Conference in Heidelberg, Germany, the 2003 IEEE Conference on Systems, Man & Cybernetics in Washington DC, and Program Co-Chair for the 1999 Fusion Conference in Sunnyvale CA.

Christopher Lintz is with Comcast Corporation.



Gabriel Commeau is a software engineer who has been working for Comcast Corporation for the last 5 years on BigData projects and real-time processing platforms. He is interested in cutting-edge technologies, especially in the domains of distributed systems, artificial intelligence and security. He currently provides technical leadership for an initiative to create an enterprise Stream Data Platform in order to manage the wide range of data streams generated by Comcast's systems.

Franchesco Dorigo is with Comcast Corporation.

Jim Fahrny is with Comcast Corporation.

Space Based Sensor Bias Estimation in the Presence of Data Association Uncertainty

DJEDJIGA BELFADEL
RICHARD W. OSBORNE, III
YAAKOV BAR-SHALOM
KRISHNA PATTIPATI

In this paper, an approach to bias estimation in the presence of measurement association uncertainty using common targets of opportunity, is developed. Data association is carried out before the estimation of sensor angle measurement biases. Consequently, the quality of data association is critical to the overall tracking performance. Data association becomes especially challenging if the sensors are passive. Mathematically, the problem can be formulated as a multidimensional optimization problem, where the objective is to maximize the generalized likelihood that the associated measurements correspond to common targets, based on target locations and sensor bias estimates. Applying gating techniques significantly reduces the size of this problem. The association likelihoods are evaluated using an exhaustive search after which an acceptance test is applied to each solution in order to obtain the correct solution. We demonstrate the merits of this approach by applying it to a simulated tracking system, which consists of two or three satellites tracking a ballistic target. We assume the sensors are synchronized, their locations are known, and we estimate their orientation biases together with the unknown target locations.

Manuscript received August 26, 2015; released for publication May 5, 2016.

Refereeing of this contribution was handled by Ramona Georgescu.

Authors' address: Electrical and Computer Engineering, University of Connecticut, Storrs, CT, U.S.A. (E-mail: {dbelfadel, rosborne, ybs, krishna}@enr.uconn.edu).

1557-6418/17/\$17.00 © 2017 JAIF

I. INTRODUCTION

Data association is a crucial task in many surveillance systems, and becomes especially challenging if the sensors are passive and measure Line of Sight (LOS) angles only for the targets. Measurements from multiple sensors have to be associated to determine the biases of the sensors and the positions of the targets from which the measurements originated. In general, the goal of data association is to partition the set of measurements across sensors into a number of subsets, in which the measurements are either from the same target (i.e., having the identical origin) or false alarms. For angle-only sensors, imperfect registration leads to LOS angle measurement errors in azimuth and elevation that can be much larger than those due to measurement noise. If uncorrected, registration errors can lead to large tracking errors and potentially to the formation of multiple tracks (ghosts) on the same target [8].

Mathematically, the problem can be formulated as a multidimensional optimization problem where the objective is to maximize the generalized likelihood, based on target locations and sensor bias estimates, that the associations correspond to real targets. Any feasible solution of this problem corresponds to a potential association hypothesis. In [14], the problem was formulated as a multidimensional assignment (S - D) problem where the objective was to maximize the likelihood that the associations correspond to targets. For $S \geq 3$, the multidimensional assignment problem is NP-hard. Many sub-optimal algorithms have been proposed to find an approximate solution, such as Lagrangian relaxation [11], greedy rounding adaptive search (GRASP) [15], genetic algorithms [3] and linear relaxation and rounding techniques [16]. Moreover, in many cases, it is possible to resort to gating techniques [10] which drastically reduce the number of decisions variables and make it possible to solve the problem optimally.

Even if a large part of the literature is devoted to this aspect, solving efficiently the multidimensional assignment problem is not the only challenge for data association problems. Indeed, the quality of near-optimal, or even optimal, solution may vary considerably depending on the context. In sparse configurations or with highly accurate sensors, the model behaves well and the optimal, or even an approximate solution, often has an acceptable percentage of correct associations. On the other hand, in medium or high density configurations or with sensors of low accuracy, the model behaves poorly, namely, there is ambiguity due to similarity of likelihoods. The optimal solution can have a poor association correctness while the correct solution can be suboptimal.

The optimal solution of the problem is supposed to be the most likely solution. As the complexity of the observed situations increases, the number of ambiguous elementary associations increases also. Since such associations get a high likelihood within the model, it usually happens that more than one solution can get

an overall likelihood very close to the likelihood of the optimal solution. In such cases, any of these solutions, including the optimal one, could appear to be the correct association hypothesis. Therefore, it seems more reasonable to consider several candidate solutions rather than by selecting only one solution, even if it has a slightly better likelihood. The general scheme underlying our approach is based on the idea of selecting several good candidate solutions, by evaluating the likelihoods, and using a goodness of fit test to obtain the correct association hypothesis.

Space-based sensors can expand the range and effectiveness of the capabilities of a Ballistic Missile Defense System (BMDS) to counter future projected threats. Integration of space based sensors into the BMDS allows for detection and tracking of threats over a larger area than ground based sensors [1]. The Space Tracking and Surveillance System (STSS) constellation consists of two or more satellites (on known trajectories) for tracking ballistic targets. Each satellite is equipped with an IR sensor that provides the azimuth and elevation to the target. The tracking problem is made more difficult due to a constant or slowly varying bias error present in each sensor's line of sight measurements.

Maximum a posteriori (MAP) data association for concurrent bias estimation and data association based on sensor-level track state estimates was proposed in [12] and extended in [13]. Sensor calibration using in-situ celestial observations to estimate bias in space-based missile tracking was proposed in [9].

In [7] we investigated the use of the minimum possible number of moving optical sensors (three or two optical sensors to observe three or six points, respectively, on the trajectory of a single target of opportunity), under the assumption of perfect data association. In the present paper, bias estimation is investigated, in the presence of false alarms, when only targets of opportunity are available. The present problem is not amenable to the multi-dimensional assignment (*S-D*, [9]) because the number of measurements needed to obtain a solution for the sensor biases presents the sequential use of 2-D assignment and relaxation as in the *S-D* algorithm, i.e., in problems where *S-D* assignment can be used one has a first solution using the first 2 lists and then, using relaxation, the remaining lists are incorporated one at a time. In the present problem the minimum number of measurements needed for a solution is as given in equation (22) and these measurements have to be correctly associated: otherwise the residual yields "unacceptable" result. Consequently one has to find such a "correct set." After this, if one uses additional measurements from the same sensors, they have to form a set of common origin (an "extra" target point), which introduces another 3 unknowns. Thus one has to find one measurement from each of (at least) two sensors (4 scalars that add 4 equations) and a search is needed until a first such set is formed (based on the residual). Then one can proceed iteratively in this fashion by adding a measurement from another sensor or a set of 2 measurements from the same 2 sensors.

For the problem considered we found that it is faster to obtain the results using directly an exhaustive search for the target points. By generating (enumerating) the set of all possible associations, which is guaranteed to contain the desired (correct association) solution, based on the association likelihoods using the target location estimates and the sensor bias estimates, an acceptance test can be applied to each solution in order to obtain the correct solution. It appears, that through the use of gating technique, the solution is obtained in a reasonable time.

We demonstrate the merits of this approach by applying it to a simulated tracking system, which consists of two or three satellites tracking a ballistic target. We assume the sensors are synchronized, their locations are known, and we estimate their orientation biases. We investigate the use of the minimum possible number of space-based sensors (which can not be less than two). Two cases are considered. In the first case, we use three optical sensors to estimate three points on the (unknown) trajectory of a single target of opportunity simultaneously with the biases of the three optical sensors [5]. In the second case, we estimate the position of six points on the trajectory of a single target of opportunity simultaneously with the biases of two space-based optical sensors [4].

Section II presents the problem formulation and solution in detail. Section III describes the simulations performed and gives the results. Finally, Section IV gives the conclusions.

II. PROBLEM FORMULATION

Assume there are N_s synchronized moving passive sensors, with known positions in the Earth Centred Inertial (ECI) Coordinate System at times t_i ,

$$\xi_s(t_i) = [\xi_s(t_i), \eta_s(t_i), \zeta_s(t_i)]', \quad s = 1, 2, \dots, N_s \quad (1)$$

and N_t target locations (target trajectory at N_t time instants of a single target) at

$$\mathbf{x}(t_i) = [x(t_i), y(t_i), z(t_i)]' \quad i = 1, 2, \dots, N_t \quad (2)$$

also in ECI coordinates. We assume that each sensor sees all the target locations (same physical target at different times).¹

The rotation between the ECI and a sensor frame is described by $\phi_s + \phi_s^n$, $\rho_s + \rho_s^n$, $\psi_s + \psi_s^n$ of sensor s as roll, pitch, and yaw respectively, where ϕ_s^n is the nominal roll angle, ϕ_s is the roll bias, etc. Each angle defines a rotation about a prescribed axis, in order to align the sensor frame axes with the ECI axes. The *xyz* rotation sequence is chosen, which is accomplished by first rotating about the *x* axis by ϕ_s^n , then rotating about the *y* axis by ρ_s^n , and finally rotating about the *z* axis by ψ_s^n . The operations needed to transform the position of a given target location at t_i expressed in ECI

¹This can also be different targets at a common time or at different times, as long as the sensors are synchronized.

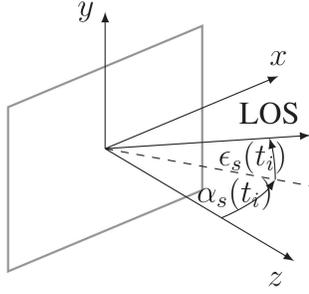


Fig. 1. Optical sensor coordinate system with the origin in the center of the focal plane.

coordinates into the sensor s coordinate system (based on its nominal orientation) is

$$\mathbf{x}_s^n(t_i) = T(\boldsymbol{\omega}_s(t_i))(\mathbf{x}(t_i) - \boldsymbol{\xi}_s(t_i))$$

$$i = 1, 2, \dots, N_t, \quad s = 1, 2, \dots, N_s \quad (3)$$

where $\boldsymbol{\omega}_s(t_i) = [\phi_s^n(t_i), \rho_s^n(t_i), \psi_s^n(t_i)]'$ is the nominal orientation of sensor s at times t_i , $T(\boldsymbol{\omega}_s(t_i))$ is the appropriate rotation matrix, and the translation $(\mathbf{x}(t_i) - \boldsymbol{\xi}_s(t_i))$ is the difference between the vector position of the target at time t_i and the vector position of the sensor s at time t_i , both expressed in ECI coordinates. The superscript “n” in (3) indicates that the rotation matrix is based on the nominal sensor orientation.

As shown in Figure 1, the azimuth angle $\alpha_s(t_i)$ is the angle in the sensor’s xz plane between the sensor’s z axis and the projection of the line of sight to the target onto the xz plane, while the elevation angle $\epsilon_s(t_i)$ is the angle between the line of sight to the target and its projection onto the xz plane, i.e.,

$$\begin{bmatrix} \alpha_s(t_i) \\ \epsilon_s(t_i) \end{bmatrix} = \begin{bmatrix} \tan^{-1} \left(\frac{x_s(t_i)}{z_s(t_i)} \right) \\ \tan^{-1} \left(\frac{y_s(t_i)}{\sqrt{x_s^2(t_i) + z_s^2(t_i)}} \right) \end{bmatrix} \quad (4)$$

The model for the biased noise-free LOS measurements is then

$$\begin{bmatrix} \alpha_s^b(t_i) \\ \epsilon_s^b(t_i) \end{bmatrix} = \begin{bmatrix} g_1(\mathbf{x}(t_i), \boldsymbol{\xi}_s(t_i), \boldsymbol{\omega}_s(t_i), \mathbf{b}_s) \\ g_2(\mathbf{x}(t_i), \boldsymbol{\xi}_s(t_i), \boldsymbol{\omega}_s(t_i), \mathbf{b}_s) \end{bmatrix}$$

$$\triangleq \mathbf{g}[\mathbf{x}(t_i), \boldsymbol{\xi}_s(t_i), \boldsymbol{\omega}_s(t_i), \mathbf{b}_s] \quad (5)$$

where g_1 and g_2 denote the sensor Cartesian coordinates-to-azimuth/elevation angle mapping that can be found by inserting (3) and (4) into (5), and the bias vector of sensor s is

$$\mathbf{b}_s = [\phi_s, \rho_s, \psi_s]'$$

$$(6)$$

For a given target, each sensor provides the noisy LOS measurements

$$\mathbf{z}_s(t_i) = \mathbf{g}[\mathbf{x}(t_i), \boldsymbol{\xi}_s(t_i), \boldsymbol{\omega}_s(t_i), \mathbf{b}_s] + \mathbf{w}_s(t_i) \quad (7)$$

where

$$\mathbf{w}_s(t_i) = [w_s^\alpha(t_i), w_s^\epsilon(t_i)]' \quad (8)$$

The measurement noises $\mathbf{w}_s(t_i)$ are zero-mean, white Gaussian with

$$R_s = \begin{bmatrix} (\sigma_s^\alpha)^2 & 0 \\ 0 & (\sigma_s^\epsilon)^2 \end{bmatrix} \quad (9)$$

and are assumed mutually independent. We shall assume, for simplicity, $\sigma_s^\alpha = \sigma_s^\epsilon = \sigma$.

The problem is to estimate the bias vectors for all sensors and the locations of the targets of opportunity. We shall obtain the maximum likelihood (ML) estimate of the augmented parameter vector

$$\boldsymbol{\theta} = [\mathbf{x}(t_1)', \dots, \mathbf{x}(t_{N_t})', \mathbf{b}_1', \dots, \mathbf{b}_{N_s}]' \quad (10)$$

consisting of the (unknown) target locations and sensor biases, by maximizing the likelihood function (LF) of $\boldsymbol{\theta}$.

III. SOLUTION

It will be assumed that there is a single target at different (unknown) locations (2), observed at times t_i , $i = 1, \dots, N_t$. The set of measurements from sensor s at time t_i is

$$Z_s(t_i) = \{\mathbf{z}_s(l, t_i)\}_{l=1}^{n_{s,i}} \quad i = 1, 2, \dots, N_t, \quad s = 1, 2, \dots, N_s \quad (11)$$

and it contains the noisy measurement from the target and clutter points or false alarms (assumed to be spatially and temporally white); the total number of measurements at sensor s at time t_i is denoted as $n_{s,i}$. The problem consists of selecting the measurement $l_{s,i}$ deemed from the target, i.e., one from each of the $N_s N_t$ lists. Due to the high accuracy of the IR spaced based sensors, we assume that each target is detected by the sensors at any given time t_i , i.e., the probability of detection $P_D = 1$. The likelihood function (LF) of $\boldsymbol{\theta}$ for a particular set of selected measurements (one from each sensor s and time t_i) assumed target-originated

$$\mathcal{L} = \{l_{s,i}\} \quad (12)$$

based on the entire set of measurements

$$\mathbf{Z} = \{Z_s(t_i) \quad i = 1, 2, \dots, N_t, \quad s = 1, 2, \dots, N_s\} \quad (13)$$

is

$$\Lambda(\boldsymbol{\theta}; \mathcal{L}, \mathbf{Z}_{\mathcal{L}}) = \prod_{i=1}^{N_t} \prod_{s=1}^{N_s} p(\mathbf{z}_s(l_{s,i}, t_i) | \boldsymbol{\theta}) \quad (14)$$

where $\mathbf{Z}_{\mathcal{L}}$ is the set of selected measurements, and

$$p[\mathbf{z}_s(l_{s,i}, t_i) | \boldsymbol{\theta}] = \prod_{i=1}^{N_t} \prod_{s=1}^{N_s} \mathcal{N}(\mathbf{z}_s(l_{s,i}, t_i); \mathbf{h}_{si}(\boldsymbol{\theta}), R_s) \quad (15)$$

and we use the compact notation

$$\mathbf{h}_{si}(\boldsymbol{\theta}) \triangleq \mathbf{g}(\mathbf{x}(t_i), \boldsymbol{\xi}_s(t_i), \boldsymbol{\omega}_s(t_i), \mathbf{b}_s) \quad (16)$$

Note that each \mathcal{L} consists of an $N_s N_t$ -tuple. The ML estimate of $\boldsymbol{\theta}$ for a certain \mathcal{L} is

$$\hat{\boldsymbol{\theta}}^{\text{ML}}(\mathcal{L}) = \arg \max_{\boldsymbol{\theta}} \Lambda(\boldsymbol{\theta}; \mathcal{L}, \mathbf{Z}_{\mathcal{L}}) \quad (17)$$

and

$$\hat{\theta}^{\text{ML}} = \hat{\theta}^{\text{ML}}(\mathcal{L}^{\text{ML}}) \quad (18)$$

where

$$\mathcal{L}^{\text{ML}} = \arg \max_{\mathcal{L}} \Lambda(\hat{\theta}^{\text{ML}}(\mathcal{L}); \mathcal{L}, Z_{\mathcal{L}}) \quad (19)$$

i.e., the final estimate (18) of (10) is based on the most likely assignment (19). The final (generalized) likelihood to be used for acceptance testing is

$$\begin{aligned} \hat{\Lambda}(\mathcal{L}) &= \Lambda(\hat{\theta}^{\text{ML}}(\mathcal{L}^{\text{ML}}); \mathcal{L}^{\text{ML}}, Z_{\mathcal{L}^{\text{ML}}}) \\ &= \prod_{i=1}^{N_t} \prod_{s=1}^{N_s} \mathcal{N}(\mathbf{z}_s(l_{s,i}, t_i); \mathbf{h}_s[\hat{\theta}^{\text{ML}}(\mathcal{L}^{\text{ML}}), t_i], R_s) \end{aligned} \quad (20)$$

Solving (17) amounts to a nonlinear LS (NLS) problem. While there are many methods to obtain $\hat{\theta}$, the iterated least squares (ILS) technique is preferred since it is easy to implement (no Hessian involved) and provides an (approximate) covariance matrix for its estimate at the same time. In order to find the MLE, one has to solve a nonlinear least squares problem for the exponent in (15). This will be done using a numerical search via the ILS technique [2].

A. Gating Region (Validation Region)

Validation gates are set up for selecting the candidate measurements originated from the target with high probability for each t_i . Measurements outside the validation regions can be ignored reasonably because the probabilities of them being from the corresponding target are quite low according to the true measurement statistical characterization. After enumerating the set of all possible associations, i.e., generating all full tuples (of length N_s) with one measurement from each of the N_s lists, the *maximum cross range error* is used in gating to prune unlikely associations. If a candidate association fails in the gating test, there is no need to use it in the likelihood cost. The calculation of the gate is recursive. Beginning with the measurement $\mathbf{z}_1(l_{1,i}, t_i)$ from the first sensor (list), we take one measurement from each list at time t_i . If the measurement from the second list $\mathbf{z}_2(l_{2,i}, t_i)$ falls inside the gate bounded by the cone with angle $(4\sigma + \max \text{ bias})$, around the $\mathbf{z}_1(l_{1,i}, t_i)$, this measurement is incorporated in the tuple for time t_i , which advances to the next list. Only full tuples (consisting of N_s LOS measurements), are to be considered. If no measurement of a particular sensor appears in any validated tuple at t_i , then none of these tuples carry information about the biases of this sensor. Consequently, none of these tuples (from t_i) will be used in the estimation of the N_s sensor biases. This is repeated for each t_i and then (16) can be carried out. Consequently, the CPU time spent in the cost computation can be reduced via the gating process.

B. Number of Hypotheses

The total number of hypotheses (combinations) for a scenario of N_t target locations and N_s sensors (assuming no missed detections) is

$$N_H = \prod_{i=1}^{N_t} \prod_{s=1}^{N_s} n_{s,i} \quad (21)$$

For example, in the case of the 2 sensors and 6 target locations, with medium clutter density, in a particular run, assume $n_{s,i}$ (number of clutter points plus the measurement from the target) as: 2, 1, 2, 1, 3, 3 for $s = 1$ and 1, 5, 2, 2, 1, 2 for $s = 2$; then the total number of hypotheses is 1440. The size of the search problem can be reduced considerably by applying gating in order to prevent implausible associations. In the previous example, only, 14% (201) passed the gating: then, this problem can be solved exactly by using an exhaustive search of modest size.

C. Requirements for bias estimability

First requirement for bias estimability. For a given target location we have a two-dimensional measurement from each sensor (the two LOS angles to the target). We assume that each sensor sees all the target locations at common times. Stacking together each measurement of N_t target locations seen by N_s sensors results in an overall measurement vector of dimension $2N_tN_s$. Given that the position and bias vectors of each target are three-dimensional, and knowing that the number of equations (size of the stacked measurement vector) has to be at least equal to the number of parameters to be estimated (target locations and biases), we must have

$$2N_tN_s \geq 3(N_t + N_s) \quad (22)$$

This is a necessary condition but not sufficient because (18) has to have a unique solution, i.e., the parameter vector has to be estimable. This is guaranteed by the second requirement.

Second requirement of bias estimability. This is the invertibility of the Fisher Information Matrix (FIM). In order to have parameter observability, the FIM must be invertible. If the FIM is not invertible (i.e., it is singular), then the CRLB (the inverse of the FIM) will not exist—the FIM will have one or more infinite eigenvalues, which means total uncertainty in a subspace of the parameter space, i.e., ambiguity [2].

For the examples of bias estimability discussed in the sequel, to estimate the biases of 3 sensors (9 bias components) we need 3 target locations (9 position components), i.e., the search is in an 18-dimensional space, while for 2 sensors (6 bias components) we need at least 6 target locations (18 position components) in order to meet the necessary requirement (22). As stated previously, the FIM must be invertible, so the rank of the FIM has to be equal to the number of parameters to be estimated (9 + 9 = 18, or 6 + 18 = 24, in the previous examples). The full rank of the FIM is a necessary and sufficient condition for estimability.

D. Iterated Least Squares for maximization of the LF of θ

Given the estimate $\hat{\theta}^j$ after j iterations, the ILS estimate after the $(j + 1)$ th iteration will be

$$\hat{\theta}^{j+1} = \hat{\theta}^j + [(H^j)'R^{-1}H^j]^{-1}(H^j)'R^{-1}[\mathbf{z} - \mathbf{h}(\hat{\theta}^j)] \quad (23)$$

where

$$\mathbf{z} = [z_1(t_1)', \dots, z_s(t_1)', \dots, z_s(t_i)', \dots, z_{N_s}(t_{N_s})']' \quad (24)$$

$$\mathbf{h}(\hat{\boldsymbol{\theta}}^j) = [h_{11}(\hat{\boldsymbol{\theta}}^j)', \dots, h_{is}(\hat{\boldsymbol{\theta}}^j)', \dots, h_{N_s N_s}(\hat{\boldsymbol{\theta}}^j)'] \quad (25)$$

$$R = \begin{bmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & R_{N_s} \end{bmatrix} \quad (26)$$

where R_s is the measurement noise covariance matrix of sensor s , and

$$H^j = \left. \frac{\partial \mathbf{h}(\boldsymbol{\theta}^j)}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^j} \quad (27)$$

is the Jacobian matrix of the vector consisting of the stacked measurement functions (25) w.r.t. (10) evaluated at the ILS estimate from the previous iteration j . In this case, the Jacobian matrix is, with the iteration index omitted for conciseness,

$$H = [H_{11} \quad H_{21} \cdots H_{N_1} \quad H_{12} \cdots H_{N_s N_s}]' \quad (28)$$

where

$$H_{is} = \begin{bmatrix} \frac{\partial g_{1_s}(t_i)}{\partial x(t_1)} & \frac{\partial g_{2_s}(t_i)}{\partial x(t_1)} \\ \frac{\partial g_{1_s}(t_i)}{\partial y(t_1)} & \frac{\partial g_{2_s}(t_i)}{\partial y(t_1)} \\ \frac{\partial g_{1_s}(t_i)}{\partial z(t_1)} & \frac{\partial g_{2_s}(t_i)}{\partial z(t_1)} \\ \vdots & \vdots \\ \frac{\partial g_{1_s}(t_i)}{\partial x(t_{N_i})} & \frac{\partial g_{2_s}(t_i)}{\partial x(t_{N_i})} \\ \frac{\partial g_{1_s}(t_i)}{\partial y(t_{N_i})} & \frac{\partial g_{2_s}(t_i)}{\partial y(t_{N_i})} \\ \frac{\partial g_{1_s}(t_i)}{\partial z(t_{N_i})} & \frac{\partial g_{2_s}(t_i)}{\partial z(t_{N_i})} \\ \frac{\partial g_{1_s}(t_i)}{\partial \psi_1} & \frac{\partial g_{2_s}(t_i)}{\partial \psi_1} \\ \frac{\partial g_{1_s}(t_i)}{\partial \rho_1} & \frac{\partial g_{2_s}(t_i)}{\partial \rho_1} \\ \frac{\partial g_{1_s}(t_i)}{\partial \phi_1} & \frac{\partial g_{2_s}(t_i)}{\partial \phi_1} \\ \vdots & \vdots \\ \frac{\partial g_{1_s}(t_i)}{\partial \psi_{N_s}} & \frac{\partial g_{2_s}(t_i)}{\partial \psi_{N_s}} \\ \frac{\partial g_{1_s}(t_i)}{\partial \rho_{N_s}} & \frac{\partial g_{2_s}(t_i)}{\partial \rho_{N_s}} \\ \frac{\partial g_{1_s}(t_i)}{\partial \phi_{N_s}} & \frac{\partial g_{2_s}(t_i)}{\partial \phi_{N_s}} \end{bmatrix} \quad (29)$$

The appropriate partial derivatives are given in the appendix.

E. Initialization

In order to perform the numerical search via ILS, an initial estimate $\hat{\boldsymbol{\theta}}^0$ is required. Assuming that the biases are null, the LOS measurements from the first and the second sensor $\alpha_1(t_i)$, $\alpha_2(t_i)$ and $\epsilon_1(t_i)$ can be used to solve for each initial Cartesian target position, in ECI coordinates, using (30)–(32).

$$x(t_i)^0 = \frac{\xi_2(t_i) - \xi_1(t_i) + \zeta_1(t_i) \tan \alpha_1(t_i) - \zeta_2(t_i) \tan \alpha_2(t_i)}{\tan \alpha_1(t_i) - \tan \alpha_2(t_i)} \quad (30)$$

$$y(t_i)^0 = \frac{\tan \alpha_1(t_i)(\xi_2(t_i) + \tan \alpha_2(t_i)(\zeta_1(t_i) - \zeta_2(t_i))) - \xi_1(t_i) \tan \alpha_2(t_i)}{\tan \alpha_1(t_i) - \tan \alpha_2(t_i)} \quad (31)$$

$$z(t_i)^0 = \eta_1(t_i) + \tan \epsilon_1(t_i) \left| \frac{(\xi_1(t_i) - \xi_2(t_i)) \cos \alpha_2(t_i) + (\zeta_2(t_i) - \zeta_1(t_i)) \sin \alpha_2(t_i)}{\sin(\alpha_1(t_i) - \alpha_2(t_i))} \right| \quad (32)$$

F. Cramér-Rao Lower Bound

In order to evaluate the efficiency of the estimator, the CRLB must be calculated. The CRLB provides a lower bound on the covariance matrix of an unbiased estimator as [1]

$$E\{(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})'\} \geq J(\boldsymbol{\theta})^{-1} \quad (33)$$

where J is the Fisher Information Matrix (FIM), $\boldsymbol{\theta}$ is the true parameter vector to be estimated, and $\hat{\boldsymbol{\theta}}$ is the estimate. The FIM is

$$J(\boldsymbol{\theta}) = E\{[\nabla_{\boldsymbol{\theta}} \ln \Lambda(\boldsymbol{\theta})][\nabla_{\boldsymbol{\theta}} \ln \Lambda(\boldsymbol{\theta})]'\} |_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{true}}} \quad (34)$$

where the gradient of the log-likelihood function is

$$\lambda(\boldsymbol{\theta}) \triangleq \ln \Lambda(\boldsymbol{\theta}) \quad (35)$$

$$\nabla_{\boldsymbol{\theta}} \lambda(\boldsymbol{\theta}) = \sum_{i=1}^{N_i} \sum_{s=1}^{N_s} H'_{is} R_s^{-1} (\mathbf{z}_s(t_i) - \mathbf{h}_{si}(\boldsymbol{\theta})) \quad (36)$$

which, when plugged into (34), gives

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{N_i} \sum_{s=1}^{N_s} H'_{is} (R_s^{-1}) H_{is} |_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{true}}} = H'(R^{-1})H |_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{true}}} \quad (37)$$

IV. SIMULATIONS

We simulate a space based system tracking a ballistic missile. The missile and satellite trajectories are generated using System Tool Kit (STK).² The target modeled represents a ballistic missile with a flight time of about 20 minutes. STK provides the target and sensor positions in three dimensional Cartesian coordinates at 1 s intervals. The target launch time is chosen so that the

²STK Systems Tool Kit are registered trademarks of Analytical Graphics, Inc.

satellite based sensors were able to follow the missile trajectory throughout its flight path.

Any association $N_S N_T$ -tuple that passes the gating test, falls into one of the following three categories:

- Completely correct (CC) association: The measurements in an association tuple have identical origin and there is no clutter measurement associated.
- Partially correct (PC) association: There are at least 2 measurements with common origin, and the rest may be from different origins or clutter measurements.
- Completely incorrect (CI) association: In an association tuple, there does not exist a pair of measurements that come from the same origin.

A. Statistical Acceptance test (Goodness of Fit)

In order to obtain the correct association, the Sum of the Normalized Square Residuals (SNSR) is used as a measure of the goodness of fit, which is defined as the minimized value of the log likelihood function (20), multiplied by 2 for convenience

$$\lambda^*(\hat{\theta}^{\text{ML}}(\mathcal{L}^{\text{ML}})) = \sum_{i=1}^{N_t} \sum_{s=1}^{N_s} \left([\mathbf{z}_s(l_{s,i}, t_i) - \mathbf{h}_{si}(\hat{\theta}^{\text{ML}}(\mathcal{L}^{\text{ML}}))] \right)' \times \mathbf{R}_s^{-1} [\mathbf{z}_s(l_{s,i}, t_i) - \mathbf{h}_{si}(\hat{\theta}^{\text{ML}}(\mathcal{L}^{\text{ML}}))] \quad (38)$$

This is similar to the linear least squares case (LS), under the Gaussian noise assumptions, where the fitting error was shown to be Chi-square distributed in [2].

In the present nonlinear LS problem, a Monte Carlo simulation is used to confirm the validity of this result, by summing up the fitting errors from N runs with independent random variables, with n_z being the number of measurements and n_x is the number of parameters, the total error obtained is Chi-square distributed with $N(n_z - n_x)$ degrees of freedom.

For the three sensor case ($n_x = 18$), the sample average SNSR over 100 Monte Carlo runs was evaluated using $n_z = 24$ LOS measurements yielding 5.71. The 99% upper limit of the probability region is, based on the $100(n_z - n_x) = 600$ degrees of freedom Chi-square distribution (divided by 100), approximately 6.83. Similar results were obtained for the two sensor case ($n_x = 24$): the sample average SNSR over 100 Monte Carlo runs was evaluated using $n_z = 28$ LOS measurements yielding 4.13. The 99% upper limit of the probability region is, based on the $100(n_z - n_x) = 400$ degrees of freedom Chi-square distribution (divided by 100), approximately 4.68.

The statistical acceptance test of an association, in a particular run, is based on data from single run,

which can be used with real data, and does not require knowledge of the true parameter. Then

$$\lambda^*(\hat{\theta}^{\text{ML}}(\mathcal{L}^{\text{ML}})) \sim \chi_{n_z - n_x}^2 \quad (39)$$

Namely, λ^* should be, with 99% probability, below the threshold $\chi_{n_z - n_x}^2(0.01)$ denoted as τ . Given an association tuple, if its SNSR (38) is less than the threshold τ , then this association is accepted, otherwise it is rejected.

For the three sensor case ($n_x = 18$), three scenarios are considered, in the first scenario, the SNSR is evaluated using $n_z = 30$ LOS measurements. The 99% upper limit of the probability region is 26.6, based on the $n_z - n_x = 12$ degrees of freedom Chi-square distribution ($\tau = 26.6$). In the second scenario, the SNSR is evaluated using $n_z = 24$ LOS measurements. The 99% upper limit of the probability region is 16.8, based on the $n_z - n_x = 6$ degrees of freedom Chi-square distribution ($\tau = 16.8$). In the third scenario, we evaluate the SNSR using an 18 LOS measurements, in this case ($\tau = 0$). Practically, in this case one has 18 unknowns and 18 nonlinear equations.

For the two sensor case ($n_x = 24$), three scenarios are considered, in the first scenario, the SNSR is evaluated using $n_z = 32$ LOS measurements. The 99% upper limit of the probability region is 20.1, based on the $n_z - n_x = 8$ degrees of freedom Chi-square distribution ($\tau = 20.1$). In the second scenario, the SNSR is evaluated using $n_z = 28$ LOS measurements. The 99% upper limit of the probability region is 13.3, based on the $n_z - n_x = 4$ degrees of freedom Chi-square distribution ($\tau = 13.3$). In the third scenario, we evaluate the SNSR using 24 LOS measurements ($\tau = 0$).

B. Three-Sensor Case

We simulated three space based optical sensors at various known orbits observing a target at three points in time at unknown locations. In this case, an 18-dimensional parameter vector is to be estimated. Figure 2 shows each target position observed by the sensors (Figure 3 gives an image of this). All the sensors are assumed to have the same accuracy, detection probability $P_D = 1$ and the expected number of false measurements at each sensor at each time is assumed to be 3. As discussed in the previous section, the three sensor biases are roll, pitch and yaw angle offsets. The biases for each sensor were set to $0.5^\circ = 8.72$ mrad. We ran 100 Monte Carlo runs. The horizontal and vertical fields-of-view of each sensor are assumed to be 60° . The measurement noise standard deviation σ_s (identical across sensors for both azimuth and elevation measurements, $\sigma_s^\alpha = \sigma_s^\epsilon = \sigma$) was assumed to be 30 μrad .

1) *Description of the Scenarios.* The sensors are assumed to provide LOS angle measurements. We denote by ξ_1, ξ_2, ξ_3 the 3D Cartesian sensor locations, and $\mathbf{x}(t_1), \mathbf{x}(t_2), \mathbf{x}(t_3)$ the 3D Cartesian target locations (all in ECI). The three target locations were chosen from a

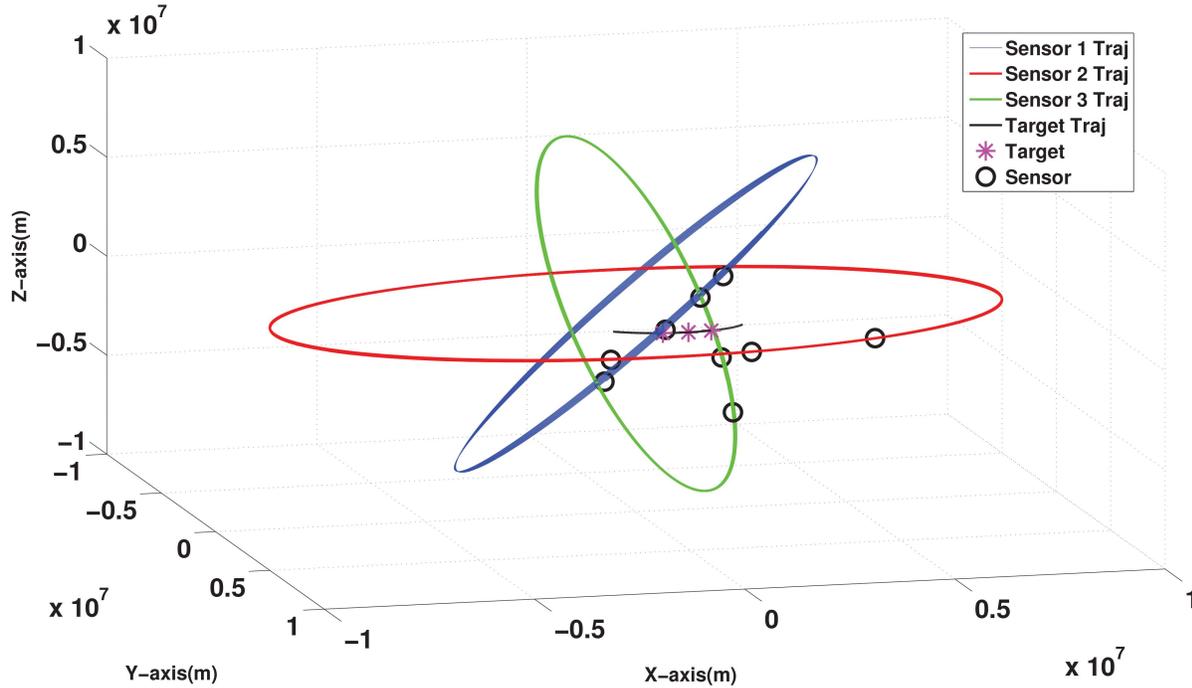


Fig. 2. Target and satellite trajectories for the three-sensor case

TABLE I
Sensor positions (km).

	ξ_1	η_1	ζ_1	ξ_2	η_2	ζ_2	ξ_3	η_3	ζ_3
Time 1	1,235	158	6,927	5,549	1,116	6,285	6,499	-279	-5,407
Time 2	1,062	-174	6,955	3,061	2,993	7,295	7,897	-719	-2,944
Time 3	887	-507	6,963	112	4,418	7,212	8,389	-1,074	-143

trajectory of a ballistic target as follows (in km)

$$\mathbf{x}(t_1) = [7,518 \quad -1,311 \quad -1,673]'$$
 (40)

$$\mathbf{x}(t_2) = [7,942 \quad -509 \quad -1,375]'$$
 (41)

$$\mathbf{x}(t_3) = [7,988 \quad 317 \quad -1,012]'$$
 (42)

Table I summarizes the sensor positions (in km).

The statistical acceptance of an association hypothesis is carried out as discussed in Sec. IV-A. The SNSR is evaluated for each validated association hypothesis. Three scenarios are considered, in the first scenario, the SNSR is evaluated using $n_z = 30$ LOS measurements. The 99% upper limit of the probability region is 26.6, based on the $n_z - n_x = 12$ degrees of freedom Chi-square distribution ($\tau = 26.6$). In the second scenario, the SNSR is evaluated using $n_z = 24$ LOS measurements. The 99% upper limit of the probability region is 16.8, based on the $n_z - n_x = 6$ degrees of freedom Chi-square distribution ($\tau = 16.8$). In the third scenario, we evaluate the SNSR using an 18 LOS measurements, in this case ($\tau = 0$). Practically, in this case one has 18 unknowns and 18 nonlinear equations and the problem is not solvable unless $P_D = 1$, in this case, we set $\tau = 0.01$ to account for numerical imprecisions. For the first scenario, the SNRS of the completely correct (CC) asso-

ciation is 5.66. The SNSR of the partially correct (PC) associations and the completely incorrect (CI) associations are of the order of 10^9 . For the second scenario, the SNSR of the completely correct (CC) association is 6.12. The SNSR of the partially correct (PC) associations and the completely incorrect (CI) associations are of the order of 10^9 . For the last scenario, the SNSR of the completely correct (CC) association is $0.23 \cdot 10^{-24}$. The SNSR of the partially correct (PC) associations and the completely incorrect (CI) associations are of the order of 10^9 .

The RMS bias errors for the correct association, are summarized in Table II, for the three scenarios in the three sensors case. The value of the σ_{CRLB} was calculated using (37) and they were provided by the ILS [6].

C. Two-Sensor Case

We simulated two space-based optical sensors at various known orbits observing a target at six (unknown) locations (which is equivalent to viewing six different targets at unknown locations). In this case, a 24-dimensional parameter vector is to be estimated. As shown in Figure 4, each target position can be observed by all sensors. All the sensors are assumed to have the

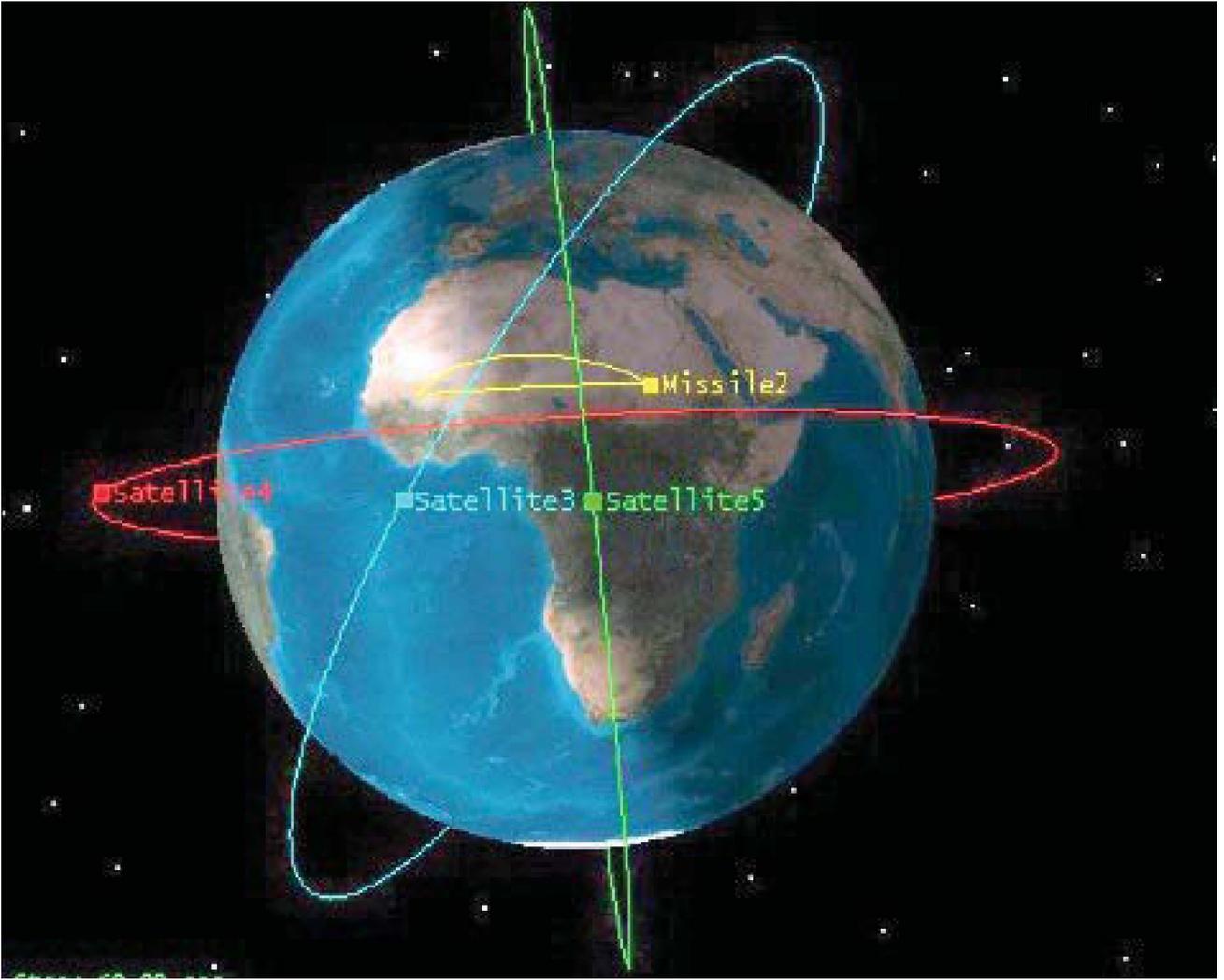


Fig. 3. Target and satellite trajectories for the three-sensor case

same accuracy, detection probability $P_D = 1$ and the expected number of false measurements at each sensor at each time is assumed to be 3. As discussed in the previous section, the three sensor biases were roll, pitch and yaw angle offsets. All the biases for each sensor were set to $0.5^\circ = 8.72$ mrad. The measurement noise standard deviation σ_s (identical across sensors for both azimuth and elevation measurements) was assumed to be $30 \mu\text{rad}$.

1) *Description of the Scenarios.* The sensors are assumed to provide LOS angle measurements. We denote by ξ_1, ξ_2 the 3D Cartesian sensor positions at six different times, and $\mathbf{x}(t_1), \mathbf{x}(t_2), \mathbf{x}(t_3), \mathbf{x}(t_4), \mathbf{x}(t_5), \mathbf{x}(t_6)$ the six 3D Cartesian target locations (all in ECI). The six target locations were chosen from a trajectory of a ballistic target as follows (in km)

$$\mathbf{x}(t_1) = [-1,167 \quad -5,782 \quad 3,028]' \quad (43)$$

$$\mathbf{x}(t_2) = [-1,054 \quad -6,027 \quad 3,436]' \quad (44)$$

$$\mathbf{x}(t_3) = [-922 \quad -6,148 \quad 3,772]' \quad (45)$$

$$\mathbf{x}(t_4) = [-774 \quad -6,155 \quad 4,036]' \quad (46)$$

$$\mathbf{x}(t_5) = [-611 \quad -6,056 \quad 4,228]' \quad (47)$$

$$\mathbf{x}(t_6) = [-435 \quad -5,852 \quad 4,344]' \quad (48)$$

Table III summarizes the sensor positions.

The statistical acceptance is done as follows. The SNSR is evaluated for each validated association hypothesis. Three scenarios were considered. In the first scenario, the SNSR is evaluated using $n_z = 32$ LOS measurements. The 99% upper limit of the probability region is 20.8, based on the 8 degrees of freedom Chi-square distribution ($\tau = 20.8$). In the second scenario, the SNSR is evaluated using $n_z = 28$ LOS measurements. The 99% upper limit of the probability region is 13.3, based on the 4 degrees of freedom Chi-square distribution ($\tau = 13.3$). In the third scenario, we evaluate the SNSR using $n_z = 24$ LOS measurements. Practically, in this case one has 24 unknowns and 24 nonlinear equations and the problem is not solvable unless $P_D = 1$, in this case, we set $\tau = 0.01$ to account for numerical imprecisions. For

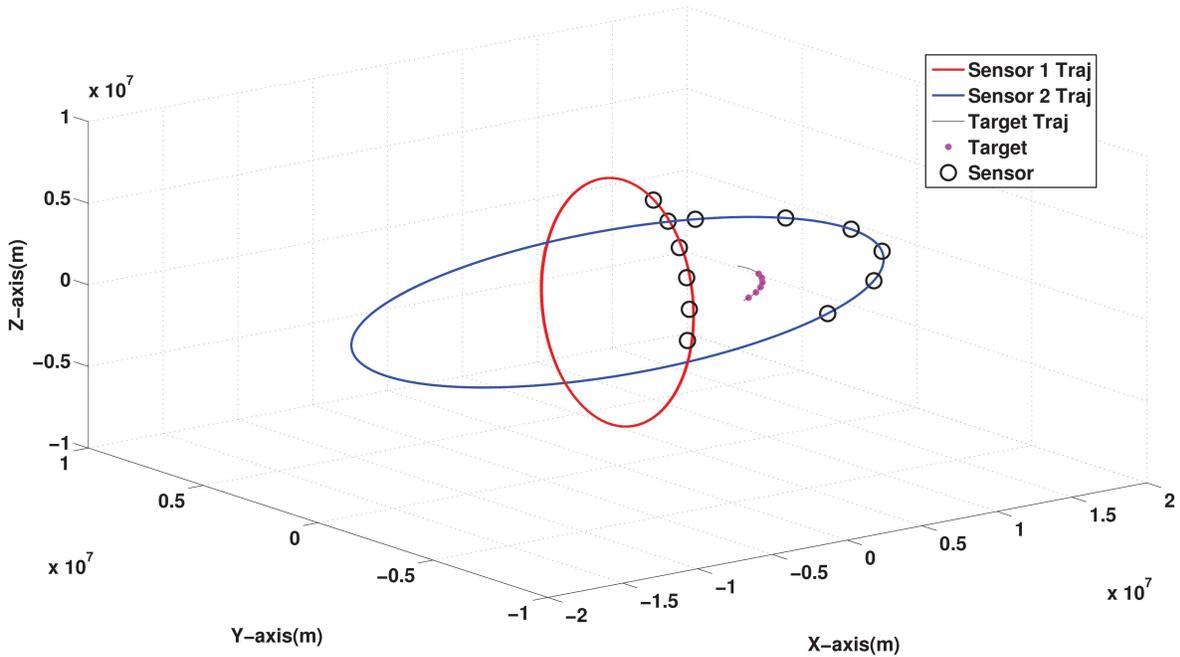


Fig. 4. Target and satellite trajectories for the two-sensor case

TABLE II
Sample average bias RMSE over 100 Monte Carlo runs and the corresponding bias standard deviation from the CRLB (σ_{CRLB})(μrad) (Three-sensor case).

Scenario		First Sensor			Second Sensor			Third Sensor		
		ψ	ρ	ϕ	ψ	ρ	ϕ	ψ	ρ	ϕ
1	RMSE	79.493	35.943	71.858	50.758	26.681	159.936	65.475	38.605	122.921
	σ_{CRLB}	78.365	39.332	85.466	50.407	25.728	152.354	69.317	38.452	133.942
2	RMSE	67.209	37.311	79.951	49.890	22.072	145.564	55.912	31.129	125.762
	σ_{CRLB}	68.909	36.620	82.351	48.584	24.235	143.217	62.641	34.364	126.637
3	RMSE	86.245	39.679	97.153	53.311	25.623	164.339	77.544	38.196	148.291
	σ_{CRLB}	78.349	39.337	85.473	50.401	25.729	152.355	69.320	38.459	133.963

the first scenario, the SNRS of the completely correct (CC) association is 6.47. The SNRS of the partially correct (PC) associations and the completely incorrect (CI) associations are of the order of 10^{10} . For the second scenario, the SNRS of the completely correct (CC) association is 7.12. The SNRS of the partially correct (PC) associations and the completely incorrect (CI) associations are of the order of 10^{10} . For the last scenario, the SNRS of the completely correct (CC) association is $0.42 \cdot 10^{-24}$. The SNRS of the partially correct (PC) associations and the completely incorrect (CI) associations are of the order of 10^{10} .

The RMS bias errors for the correct association, are summarized in Table IV, for the three scenarios in the two sensors case.

V. CONCLUSIONS

In this paper we presented an approach to bias estimation in the presence of measurement association

TABLE III
Sensor positions (km).

	ξ_1	η_1	ζ_1	ξ_2	η_2	ζ_2
t_1	187	-1,439	6,886	-3,966	-5,969	8,519
t_2	-902	-2,786	6,400	123	-7,238	8,458
t_3	-1,934	-3,951	5,494	4,195	-7,436	7,145
t_4	-2,840	-4,858	4,229	7,646	-6,533	4,774
t_5	-3,559	-5,447	2,687	9,965	-4,664	1,698
t_6	-4,046	-5,680	968	10,810	-2,105	-1,630

uncertainty using common targets of opportunity. The association likelihoods are evaluated, following gating, using an exhaustive search after which a statistical acceptance test is applied to each solution in order to discriminate the correct solution from the incorrect associations. Using simulated space based tracking systems consisting of two or three satellites tracking a ballistic target, we showed that this approach performs well.

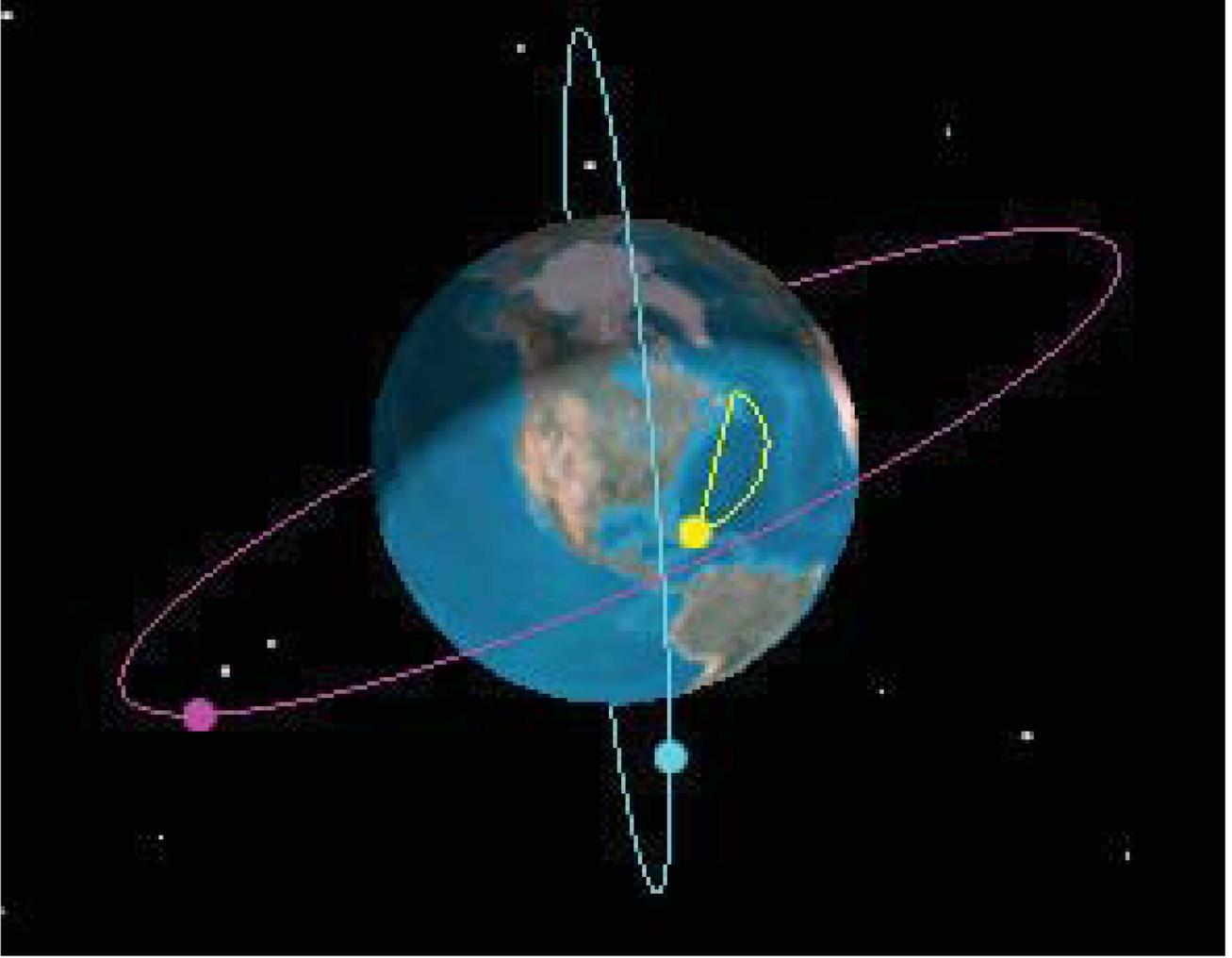


Fig. 5. Target and satellite trajectories for the two-sensor case

TABLE IV

Sample average bias RMSE over 100 Monte Carlo runs and the corresponding bias standard deviation from the CRLB (σ_{CRLB})(μrad) (Two-sensor case).

Scenario		First Sensor			Second Sensor		
		ψ	ρ	ϕ	ψ	ρ	ϕ
1	RMSE	128.469	139.761	164.244	74.097	43.693	166.525
	σ_{CRLB}	133.688	150.919	165.933	73.772	46.724	164.050
2	RMSE	143.732	148.461	173.969	80.755	49.571	173.860
	σ_{CRLB}	133.609	151.170	165.929	73.865	46.622	164.23
3	RMSE	149.383	168.707	180.788	82.082	52.476	181.479
	σ_{CRLB}	133.784	151.194	177.097	74.251	46.727	170.014

Another significance of this work is the formulation of a measure of the goodness of fit (Sum of the Normalized Square Residuals—(SNSR)) for the nonlinear least squares case, under Gaussian noise assumptions. Similarly, to the linear least squares case, where the fitting error was shown to be Chi-square distributed [2], we showed that this can be used in the nonlinear LS, thus providing a statistical test that selects the correct associations.

APPENDIX A PARTIAL DERIVATIVES

The appropriate partial derivatives of (29) are

$$\frac{\partial g_{1_s}(t_i)}{\partial x(t_k)} = \frac{\partial g_{1_s}(t_i)}{\partial x_s(t_i)} \frac{\partial x_s(t_i)}{\partial x(t_k)} + \frac{\partial g_{1_s}(t_i)}{\partial y_s(t_i)} \frac{\partial y_s(t_i)}{\partial x(t_k)} + \frac{\partial g_{1_s}(t_i)}{\partial z_s(t_i)} \frac{\partial z_s(t_i)}{\partial x(t_k)} \quad (49)$$

$$\frac{\partial g_{1_s}(t_i)}{\partial y(t_k)} = \frac{\partial g_{1_s}(t_i)}{\partial x_s(t_i)} \frac{\partial x_s(t_i)}{\partial y(t_k)} + \frac{\partial g_{1_s}(t_i)}{\partial y_s(t_i)} \frac{\partial y_s(t_i)}{\partial y(t_k)} + \frac{\partial g_{1_s}(t_i)}{\partial z_s(t_i)} \frac{\partial z_s(t_i)}{\partial y(t_k)} \quad (50)$$

$$\frac{\partial g_{1_s}(t_i)}{\partial z(t_k)} = \frac{\partial g_{1_s}(t_i)}{\partial x_s(t_i)} \frac{\partial x_s(t_i)}{\partial z(t_k)} + \frac{\partial g_{1_s}(t_i)}{\partial y_s(t_i)} \frac{\partial y_s(t_i)}{\partial z(t_k)} + \frac{\partial g_{1_s}(t_i)}{\partial z_s(t_i)} \frac{\partial z_s(t_i)}{\partial z(t_k)} \quad (51)$$

$$\frac{\partial g_{1_s}(t_i)}{\partial \psi_k} = \frac{\partial g_{1_s}(t_i)}{\partial x_s(t_i)} \frac{\partial x_s(t_i)}{\partial \psi_k} + \frac{\partial g_{1_s}(t_i)}{\partial y_s(t_i)} \frac{\partial y_s(t_i)}{\partial \psi_k} + \frac{\partial g_{1_s}(t_i)}{\partial z_s(t_i)} \frac{\partial z_s(t_i)}{\partial \psi_k} \quad (52)$$

$$\frac{\partial g_{1_s}(t_i)}{\partial \rho_k} = \frac{\partial g_{1_s}(t_i)}{\partial x_s(t_i)} \frac{\partial x_s(t_i)}{\partial \rho_k} + \frac{\partial g_{1_s}(t_i)}{\partial y_s(t_i)} \frac{\partial y_s(t_i)}{\partial \rho_k} + \frac{\partial g_{1_s}(t_i)}{\partial z_s(t_i)} \frac{\partial z_s(t_i)}{\partial \rho_k} \quad (53)$$

$$\frac{\partial g_{1_s}(t_i)}{\partial \phi_k} = \frac{\partial g_{1_s}(t_i)}{\partial x_s(t_i)} \frac{\partial x_s(t_i)}{\partial \phi_k} + \frac{\partial g_{1_s}(t_i)}{\partial y_s(t_i)} \frac{\partial y_s(t_i)}{\partial \phi_k} + \frac{\partial g_{1_s}(t_i)}{\partial z_s(t_i)} \frac{\partial z_s(t_i)}{\partial \phi_k} \quad (54)$$

$$\frac{\partial g_{2_s}(t_i)}{\partial x(t_k)} = \frac{\partial g_{2_s}(t_i)}{\partial x_s(t_i)} \frac{\partial x_s(t_i)}{\partial x(t_k)} + \frac{\partial g_{2_s}(t_i)}{\partial y_s(t_i)} \frac{\partial y_s(t_i)}{\partial x(t_k)} + \frac{\partial g_{2_s}(t_i)}{\partial z_s(t_i)} \frac{\partial z_s(t_i)}{\partial x(t_k)} \quad (55)$$

$$\frac{\partial g_{2_s}(t_i)}{\partial y(t_k)} = \frac{\partial g_{2_s}(t_i)}{\partial x_s(t_i)} \frac{\partial x_s(t_i)}{\partial y(t_k)} + \frac{\partial g_{2_s}(t_i)}{\partial y_s(t_i)} \frac{\partial y_s(t_i)}{\partial y(t_k)} + \frac{\partial g_{2_s}(t_i)}{\partial z_s(t_i)} \frac{\partial z_s(t_i)}{\partial y(t_k)} \quad (56)$$

$$\frac{\partial g_{2_s}(t_i)}{\partial z(t_k)} = \frac{\partial g_{2_s}(t_i)}{\partial x_s(t_i)} \frac{\partial x_s(t_i)}{\partial z(t_k)} + \frac{\partial g_{2_s}(t_i)}{\partial y_s(t_i)} \frac{\partial y_s(t_i)}{\partial z(t_k)} + \frac{\partial g_{2_s}(t_i)}{\partial z_s(t_i)} \frac{\partial z_s(t_i)}{\partial z(t_k)} \quad (57)$$

$$\frac{\partial g_{2_s}(t_i)}{\partial \psi_k} = \frac{\partial g_{2_s}(t_i)}{\partial x_s(t_i)} \frac{\partial x_s(t_i)}{\partial \psi_k} + \frac{\partial g_{2_s}(t_i)}{\partial y_s(t_i)} \frac{\partial y_s(t_i)}{\partial \psi_k} + \frac{\partial g_{2_s}(t_i)}{\partial z_s(t_i)} \frac{\partial z_s(t_i)}{\partial \psi_k} \quad (58)$$

$$\frac{\partial g_{2_s}(t_i)}{\partial \rho_k} = \frac{\partial g_{2_s}(t_i)}{\partial x_s(t_i)} \frac{\partial x_s(t_i)}{\partial \rho_k} + \frac{\partial g_{2_s}(t_i)}{\partial y_s(t_i)} \frac{\partial y_s(t_i)}{\partial \rho_k} + \frac{\partial g_{2_s}(t_i)}{\partial z_s(t_i)} \frac{\partial z_s(t_i)}{\partial \rho_k} \quad (59)$$

$$\frac{\partial g_{2_s}(t_i)}{\partial \phi_k} = \frac{\partial g_{2_s}(t_i)}{\partial x_s(t_i)} \frac{\partial x_s(t_i)}{\partial \phi_k} + \frac{\partial g_{2_s}(t_i)}{\partial y_s(t_i)} \frac{\partial y_s(t_i)}{\partial \phi_k} + \frac{\partial g_{2_s}(t_i)}{\partial z_s(t_i)} \frac{\partial z_s(t_i)}{\partial \phi_k} \quad (60)$$

Given that (3) can be written as

$$\begin{aligned} \mathbf{x}_s(t_i) &= \begin{bmatrix} x_s(t_i) \\ y_s(t_i) \\ z_s(t_i) \end{bmatrix} = T_s(\mathbf{x}(t_i) - \xi_s) \\ &= \begin{bmatrix} T_{s11} & T_{s12} & T_{s13} \\ T_{s21} & T_{s22} & T_{s23} \\ T_{s31} & T_{s32} & T_{s33} \end{bmatrix} \begin{bmatrix} x(t_i) - \xi_s \\ y(t_i) - \eta_s \\ z(t_i) - \zeta_s \end{bmatrix} \end{aligned} \quad (61)$$

therefore

$$x_s(t_i) = T_{s11}(x(t_i) - \xi_s) + T_{s12}(y(t_i) - \eta_s) + T_{s13}(z(t_i) - \zeta_s) \quad (62)$$

$$y_s(t_i) = T_{s21}(x(t_i) - \xi_s) + T_{s22}(y(t_i) - \eta_s) + T_{s23}(z(t_i) - \zeta_s) \quad (63)$$

$$z_s(t_i) = T_{s31}(x(t_i) - \xi_s) + T_{s32}(y(t_i) - \eta_s) + T_{s33}(z(t_i) - \zeta_s) \quad (64)$$

and

$$\begin{aligned} \frac{\partial x_s(t_i)}{\partial x(t_k)} &= T_{s11}, & \frac{\partial x_s(t_i)}{\partial y(t_k)} &= T_{s12}, & \frac{\partial x_s(t_i)}{\partial z(t_k)} &= T_{s13} \\ \frac{\partial y_s(t_i)}{\partial x(t_k)} &= T_{s21}, & \frac{\partial y_s(t_i)}{\partial y(t_k)} &= T_{s22}, & \frac{\partial y_s(t_i)}{\partial z(t_k)} &= T_{s23} \\ \frac{\partial z_s(t_i)}{\partial x(t_k)} &= T_{s31}, & \frac{\partial z_s(t_i)}{\partial y(t_k)} &= T_{s32}, & \frac{\partial z_s(t_i)}{\partial z(t_k)} &= T_{s33} \end{aligned} \quad (65)$$

$$\frac{\partial x_s(t_i)}{\partial \psi_k} = \frac{\partial T_{s11}}{\partial \psi_k}(x(t_i) - \xi_s) + \frac{\partial T_{s12}}{\partial \psi_k}(y(t_i) - \eta_s) + \frac{\partial T_{s13}}{\partial \psi_k}(z(t_i) - \zeta_s) \quad (66)$$

$$\frac{\partial x_s(t_i)}{\partial \rho_k} = \frac{\partial T_{s11}}{\partial \rho_k}(x(t_i) - \xi_s) + \frac{\partial T_{s12}}{\partial \rho_k}(y(t_i) - \eta_s) + \frac{\partial T_{s13}}{\partial \rho_k}(z(t_i) - \zeta_s) \quad (67)$$

$$\frac{\partial x_s(t_i)}{\partial \phi_k} = \frac{\partial T_{s11}}{\partial \phi_k}(x(t_i) - \xi_s) + \frac{\partial T_{s12}}{\partial \phi_k}(y(t_i) - \eta_s) + \frac{\partial T_{s13}}{\partial \phi_k}(z(t_i) - \zeta_s) \quad (68)$$

$$\frac{\partial y_s(t_i)}{\partial \psi_k} = \frac{\partial T_{s21}}{\partial \psi_k}(x(t_i) - \xi_s) + \frac{\partial T_{s22}}{\partial \psi_k}(y(t_i) - \eta_s) + \frac{\partial T_{s23}}{\partial \psi_k}(z(t_i) - \zeta_s) \quad (69)$$

$$\frac{\partial y_s(t_i)}{\partial \rho_k} = \frac{\partial T_{s21}}{\partial \rho_k}(x(t_i) - \xi_s) + \frac{\partial T_{s22}}{\partial \rho_k}(y(t_i) - \eta_s) + \frac{\partial T_{s23}}{\partial \rho_k}(z(t_i) - \zeta_s) \quad (70)$$

$$\frac{\partial y_s(t_i)}{\partial \phi_k} = \frac{\partial T_{s21}}{\partial \phi_k}(x(t_i) - \xi_s) + \frac{\partial T_{s22}}{\partial \phi_k}(y(t_i) - \eta_s) + \frac{\partial T_{s23}}{\partial \phi_k}(z(t_i) - \zeta_s) \quad (71)$$

$$\frac{\partial z_s(t_i)}{\partial \psi_k} = \frac{\partial T_{s31}}{\partial \psi_k}(x(t_i) - \xi_s) + \frac{\partial T_{s32}}{\partial \psi_k}(y(t_i) - \eta_s) + \frac{\partial T_{s33}}{\partial \psi_k}(z(t_i) - \zeta_s) \quad (72)$$

$$\frac{\partial z_s(t_i)}{\partial \rho_k} = \frac{\partial T_{s31}}{\partial \rho_k}(x(t_i) - \xi_s) + \frac{\partial T_{s32}}{\partial \rho_k}(y(t_i) - \eta_s) + \frac{\partial T_{s33}}{\partial \rho_k}(z(t_i) - \zeta_s) \quad (73)$$

$$\frac{\partial z_s(t_i)}{\partial \phi_k} = \frac{\partial T_{s31}}{\partial \phi_k}(x(t_i) - \xi_s) + \frac{\partial T_{s32}}{\partial \phi_k}(y(t_i) - \eta_s) + \frac{\partial T_{s33}}{\partial \phi_k}(z(t_i) - \zeta_s) \quad (74)$$

$$\frac{\partial g_{1_s}(t_i)}{\partial x_s(t_i)} = \frac{z_s(t_i)}{z_s(t_i)^2 + x_s(t_i)^2} \quad (75)$$

$$\frac{\partial g_{1_s}(t_i)}{\partial y_s(t_i)} = 0 \quad (76)$$

$$\frac{\partial g_{1_s}(t_i)}{\partial z_s(t_i)} = -\frac{x_s(t_i)}{x_s(t_i)^2 + z_s(t_i)^2} \quad (77)$$

$$\frac{\partial g_{2_s}(t_i)}{\partial x_s(t_i)} = -\frac{x_s(t_i)y_s(t_i)}{\sqrt{(x_s(t_i)^2 + z_s(t_i)^2)(x_s(t_i)^2 + y_s(t_i)^2 + z_s(t_i)^2)}} \quad (78)$$

$$\frac{\partial g_{2_s}(t_i)}{\partial y_s(t_i)} = \frac{\sqrt{x_s(t_i)^2 + z_s(t_i)^2}}{x_s(t_i)^2 + y_s(t_i)^2 + z_s(t_i)^2} \quad (79)$$

$$\frac{\partial g_{2_s}(t_i)}{\partial z_s(t_i)} = -\frac{z_s(t_i)y_s(t_i)}{(x_s(t_i)^2 + y_s(t_i)^2 + z_s(t_i)^2) \left(\sqrt{x_s(t_i)^2 + z_s(t_i)^2} \right)} \quad (80)$$

$$\frac{\partial T_{s11}}{\partial \psi_k} = -\sin \psi_k \cos \rho_k \quad (81)$$

$$\frac{\partial T_{s12}}{\partial \psi_k} = -\sin \psi_k \sin \rho_k \sin \phi_k - \cos \psi_k \cos \phi_k \quad (82)$$

$$\frac{\partial T_{s13}}{\partial \psi_k} = -\sin \psi_k \sin \rho_k \cos \phi_k + \cos \psi_k \sin \phi_k \quad (83)$$

$$\frac{\partial T_{s21}}{\partial \psi_k} = \cos \psi_k \cos \rho_k \quad (84)$$

$$\frac{\partial T_{s22}}{\partial \psi_k} = \cos \psi_k \sin \rho_k \sin \phi_k - \sin \psi_k \cos \phi_k \quad (85)$$

$$\frac{\partial T_{s23}}{\partial \psi_k} = \cos \psi_k \sin \rho_k \cos \phi_k + \sin \psi_k \sin \phi_k \quad (86)$$

$$\frac{\partial T_{s31}}{\partial \psi_k} = 0 \quad (87)$$

$$\frac{\partial T_{s32}}{\partial \psi_k} = 0 \quad (88)$$

$$\frac{\partial T_{s33}}{\partial \psi_k} = 0 \quad (89)$$

$$\frac{\partial T_{s11}}{\partial \rho_k} = -\cos \psi_k \sin \rho_k \quad (90)$$

$$\frac{\partial T_{s12}}{\partial \rho_k} = \cos \psi_k \cos \rho_k \sin \phi_k \quad (91)$$

$$\frac{\partial T_{s13}}{\partial \rho_k} = \cos \psi_k \cos \rho_k \cos \phi_k \quad (92)$$

$$\frac{\partial T_{s21}}{\partial \rho_k} = -\sin \psi_k \sin \phi_k \quad (93)$$

$$\frac{\partial T_{s22}}{\partial \rho_k} = \sin \psi_k \cos \rho_k \sin \phi_k \quad (94)$$

$$\frac{\partial T_{s23}}{\partial \rho_k} = \sin \psi_k \cos \rho_k \cos \phi_k \quad (95)$$

$$\frac{\partial T_{s31}}{\partial \rho_k} = -\cos \phi_k \quad (96)$$

$$\frac{\partial T_{s32}}{\partial \rho_k} = -\sin \rho_k \sin \phi_k \quad (97)$$

$$\frac{\partial T_{s33}}{\partial \rho_k} = -\sin \rho_k \cos \phi_k \quad (98)$$

$$\frac{\partial T_{s11}}{\partial \phi_k} = 0 \quad (99)$$

$$\frac{\partial T_{s12}}{\partial \phi_k} = \cos \psi_k \sin \rho_k \cos \phi_k + \sin \psi_k \sin \phi_k \quad (100)$$

$$\frac{\partial T_{s13}}{\partial \phi_k} = -\cos \psi_k \sin \rho_k \sin \phi_k + \sin \psi_k \cos \phi_k \quad (101)$$

$$\frac{\partial T_{s21}}{\partial \phi_k} = 0 \quad (102)$$

$$\frac{\partial T_{s22}}{\partial \phi_k} = \sin \psi_k \sin \rho_k \cos \phi_k - \cos \psi_k \sin \phi_k \quad (103)$$

$$\frac{\partial T_{s23}}{\partial \phi_k} = -\sin \psi_k \sin \rho_k \sin \phi_k - \cos \psi_k \cos \phi_k \quad (104)$$

$$\frac{\partial T_{s31}}{\partial \phi_k} = 0 \quad (105)$$

$$\frac{\partial T_{s32}}{\partial \phi_k} = \cos \psi_k \cos \phi_k \quad (106)$$

$$\frac{\partial T_{s33}}{\partial \phi_k} = -\cos \rho_k \sin \phi_k \quad (107)$$

REFERENCES

- [1] "FTM-20 in the Ballistic Missile Defense System," AEGIS Ballistic Missile Defense System, Tech. Rep., approved For Public Release 13-MDA-7168 (5 Feb 13).
- [2] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. J. Wiley and Sons, 2001.
- [3] M. Basseville
A genetic algorithm based multi-dimensional data association algorithm for multi-sensor multi-target tracking, *Mathematical and Computer Modelling*, vol. 26, no. 4, Dec. 1997.
- [4] D. Belfadel, R. W. Osborne, and Y. Bar-Shalom
A Minimalist Approach to Bias Estimation for Passive Sensor Measurements with Targets of Opportunity, in *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, #8857-13, San Diego, California, Aug. 2013.
- [5] D. Belfadel, R. W. Osborne, and Y. Bar-Shalom
Bias Estimation for Optical Sensor Measurements with Targets of Opportunity, in *Proc. FUSION Conf.*, Istanbul, Turkey, July 2013.
- [6] D. Belfadel, R. W. Osborne, and Y. Bar-Shalom
Bias Estimation and Observability for Optical Sensor Measurements with Targets of Opportunity, *Journal of Advances in Information Fusion*, vol. 9, no. 2, Dec. 2014.
- [7] D. Belfadel, R. W. Osborne, and Y. Bar-Shalom
Bias Estimation for Moving Optical Sensor Measurements with Targets of Opportunity, *JEIF*, vol. 24, no. 3, Dec. 2014.
- [8] D. Belfadel, R. W. Osborne, and Y. Bar-Shalom
Bias estimation for space-based optical sensors with targets of opportunity, in *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, #9092-25, Baltimore, MD, May. 2014.
- [9] T. M. Clemons and K.-C. Chang
Sensor Calibration using In-Situ Celestial Observations to Estimate Bias in Space-Based Missile Tracking, *IEEE Trans. on Aerospace and Electronic Systems*, vol. 48, no. 2, pp. 1403–1427, April 2012.
- [10] J. B. Collins and J. K. Uhlmann
Efficient gating in data association with multivariate distributed states, *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 3, 1992.
- [11] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom
A generalized S-D assignment algorithm for multisensor-multitarget state estimation, *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 2, Dec. 1997.

- [12] B. D. Kragel, S. Danford, S. M. Herman, and A. B. Poore
Joint MAP Bias Estimation and Data Association: Algorithms,
Proc. SPIE Conf. on Signal and Data Processing of Small Targets, #6699-1E, Aug. 2007.
- [13] B. D. Kragel, S. Danford, and A. B. Poore
Concurrent MAP Data Association and Absolute Bias Estimation with an Arbitrary Number of Sensors,
Proc. SPIE Conf. on Signal and Data Processing of Small Targets, #6969-50, May 2008.
- [14] K. R. Pattipati, S. Deb, Y. Bar-Shalom, and R. Washburn
A new relaxation algorithm and passive sensor data association,
IEEE Trans. Automat. Contr., vol. 37, no. 2, pp. 198–213, Apr. 1992.
- [15] R. Robertson
A set of greedy rounding randomized adaptative local search procedure (grasp) implementation for the multidimensional assignment problem,
Computational Optimization and Applications, vol. 19, Dec. 2001.
- [16] P. P. A. Storms and F. C. R. Spieksma
An LP-based algorithm for the data association problem in multitarget tracking,
Computers and Operations Research, vol. 30, no. 7, 2003.



Djedjiga Belfadel is an Assistant Professor in the Electrical and Computer Engineering department at Fairfield University, Fairfield, CT. She obtained her B.S., degrees from the University of Mouloud Mammeri in 2003, her M.S., degrees from the University of New Haven in 2008, and her Ph.D. degree from University of Connecticut in 2015, all in electrical engineering. From 2009 to 2011, she worked, as an Electrical Engineer, at Evax Systems Inc. in Branford, Connecticut. Her research interests include target tracking, data association, sensor fusion, machine vision, and other aspects of estimation.



Richard W. Osborne, III obtained his B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Connecticut in 2004, 2007, and 2012, respectively. From 2012–2014 he was an Assistant Research Professor in the Electrical Engineering department at the University of Connecticut, Storrs, CT. From 2014–2015 he was a Senior Research Engineer at BAE Systems, Inc. in Burlington, MA, and since 2015, he has been a Senior Research Engineer at United Technologies Research Center in East Hartford, CT. His academic interests include adaptive target tracking, information/sensor fusion, perception/computer vision, autonomous systems, and other aspects of estimation.

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 Dept. 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 550 papers and book chapters. He coauthored/edited 8 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 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 during 2004–13. 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. This award has been renamed in 2016 as the Yaakov Bar-Shalom Award for a Lifetime of Excellence in Information Fusion.



Krishna R. Pattipati received the B. Tech. degree in electrical engineering with highest honors from the Indian Institute of Technology, Kharagpur, in 1975, and the M.S. and Ph.D. degrees in systems engineering from UConn, Storrs, in 1977 and 1980, respectively. He was with ALPHATECH, Inc., Burlington, MA from 1980 to 1986. He has been with the department of Electrical and Computer Engineering at UConn, where he is currently the Board of Trustees Distinguished Professor and the UTC Chair Professor in Systems Engineering. Dr. Pattipati's research activities are in the areas of proactive decision support, uncertainty quantification, smart manufacturing, autonomy, knowledge representation, and optimization-based learning and inference. A common theme among these applications is that they are characterized by a great deal of uncertainty, complexity, and computational intractability. He is a cofounder of Qualtech Systems, Inc., a firm specializing in advanced integrated diagnostics software tools (TEAMS, TEAMS-RT, TEAMS-RDS, TEAMATE), and serves on the board of Aptima, Inc. Dr. Pattipati was selected by the IEEE Systems, Man, and Cybernetics (SMC) Society as the Outstanding Young Engineer of 1984, and received the Centennial Key to the Future award. He has served as the Editor-in-Chief of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B from 1998 to 2001, Vice-President for Technical Activities of the IEEE SMC Society from 1998 to 1999, and as Vice-President for Conferences and Meetings of the IEEE SMC Society from 2000 to 2001. He was co-recipient of the Andrew P. Sage Award for the Best SMC Transactions Paper for 1999, the Barry Carlton Award for the Best AES Transactions Paper for 2000, the 2002 and 2008 NASA Space Act Awards for "A Comprehensive Toolset for Model-based Health Monitoring and Diagnosis," and "Real-time Update of Fault-Test Dependencies of Dynamic Systems: A Comprehensive Toolset for Model-Based Health Monitoring and Diagnostics," and the 2003 AAUP Research Excellence Award at UCONN. He also won the best technical paper awards at the 1985, 1990, 1994, 2002, 2004, 2005 and 2011 IEEE AUTOTEST Conferences, and at the 1997, 2004 Command and Control Conference. He is an elected Fellow of IEEE and of the Connecticut Academy of Science and Engineering.



Analysis of Log-Homotopy Based Particle Flow Filters

MUHAMMAD ALTAMASH KHAN
MARTIN ULMKE
WOLFGANG KOCH

The state estimation plays an important role in analyzing many real world systems. Such systems can be classified into being linear or non-linear, and depending on the statistical properties of the inherent uncertainties as being Gaussian or non-Gaussian. Unlike linear Gaussian systems, a close form estimator does not exist for non-linear/non-Gaussian systems. Typical solutions like EKF/UKF can fail, while Monte Carlo methods even though more accurate, are computationally expensive. Recently proposed log homotopy based particle flow filters, also known as Daum-Huang filters (DHF) provide an alternative way for non-linear, non-Gaussian state estimation. There have been a number of DHF derived, based on solutions of the homotopy flow equation. The performance of these new filters depends strongly on the implementation methodology. In this paper, we study a non-linear system, perturbed by Gaussian and non-Gaussian noises. We highlight the key factors affecting the DHF performance, and investigate them individually in detail. We then make recommendations based on our results. It is shown that a properly designed DHF can outperform a basic particle filter, with less execution time.

Manuscript received December 3, 2015; released for publication October 6, 2016.

Refereeing of this contribution was handled by Ramona Georgescu.

Authors' address: Sensor Data and Information Fusion, Fraunhofer FKIE, Wachtberg, Germany (E-mail: {altamash.khan, martin.ulmke, wolfgang.koch}@fkie.fraunhofer.de).

1557-6418/17/\$17.00 © 2017 JAIF

I. INTRODUCTION

The Bayesian estimation framework offers an intuitive way for the estimation of hidden states of a dynamical system based on the observational data. The Bayesian estimation is carried out recursively, typically consisting of a prediction and a correction step. A transition density describes the time evolution of the state conditioned on the previous values, while a measurement density describes the likelihood of measurements given the current state. These densities are then used recursively for the evaluation of prior and posterior state distributions at any given moment of time. The process is known as recursive Bayesian estimation (RBE) and arises in many real scenarios. Finite dimensional analytical solutions to the RBE problem are available only in few cases, mainly when the system model is linear Gaussian (Kalman filter) or a finite state Hidden Markov model (HMM) [1]. Traditional methods for non-linear state estimation include Extended (EKF) and Unscented Kalman filter (UKF). However these methods are generally sub-optimal and their performance degrades with the increase in the non-linearity, and also when the transition and measurement densities are non-Gaussian (e.g. multimodal, exponential).

Particle filters, also known as sequential Monte Carlo (SMC) methods, provide an alternative way to the state estimation. The main idea is to represent the posterior density by a weighted set of random samples (particles), which are then used to form the point estimates, e.g., mean and variance [2]. The posterior density under these settings approximately represents the path distribution, i.e., distribution of the state through the time, conditioned on the measurements. Several version of particle filters have been proposed in the literature, e.g., sampling importance resampling (SIR) filter also known as bootstrap particle filter [3], auxiliary sampling importance resampling (ASIR) filter [4], regularized particle filter (RPF) [5] etc. While particle filters can effectively deal with the non-linearities and non-Gaussian noises, they suffer from the so called *weight degeneracy* and *curse of dimensionality*. Weight degeneracy refers to the fact that after few updates all but one particle have negligible weights. Weight degeneracy occurs when the target distribution does not significantly overlap with the prior distribution. Several solutions have been proposed to address these problem e.g. re-sampling, the use of Markov Chain Monte Carlo (MCMC) methods, use of bridging densities as suggested in [6] and [7]. Bridging densities are obtained by varying the so called *progression parameter*, which corresponds to the gradual introduction of the measurements. In this manner the posterior density can be better approximated. On the other hand, the curse of dimensionality means that to maintain a certain performance level, the required number of particles increases exponentially with the increase in the state dimension, as reported in [8].

A different approach to non-linear filtering has been suggested by Daum and Huang in a series of papers [9]–[15], which is based on the gradual inclusion of the measurements. The key idea is to model the transition of particles from the prior to the posterior density as a physical flow under the influence of an external force (measurements). Particles are sampled from the state transition density and a notion of synthetic time also called the pseudo-time is introduced, in which particles flow until they reach *correct* posterior locations. A stochastic differential equation (SDE) define the flow of particles in pseudo-time, while the Fokker-Planck equation (FPE) describes the density evolution. A flow vector is obtained by solving the FPE under different assumptions, which is then integrated numerically yielding updated states of particles. The new filter is termed as homotopy based particle flow filter or simply Daum-Huang filter (DHF) after the developers. Different flow solutions have been derived, including the incompressible flow [9], zero diffusion exact flow [10], Coulomb’s law flow [11] and zero-curvature flow [12] non zero diffusion flow [13].

DHF implementations have been reported in several publications. While conceptually being quite intuitive, DHF performance suffers in practice due to several assumptions, made both in the theory and the implementation. In this paper we identify key factors affecting the performance of the DHF. We study each of those factors in detail and in the light of the results, we suggest possible improvements in the DHF implementation. We consider state estimation of a non-linear system under both Gaussian and non-Gaussian measurement noises. The effect of different methods on the performance of DHF is studied individually for both noise cases. We show that by a careful design, the DHF performance can be substantially improved over the more traditional implementations.

The outline of the paper is given as follows: We present a description of homotopy based particle flow in section II. We start with the general formulation of RBE. We then give a derivation of the generic homotopy based flow equation, which is followed by its specific solutions. Next, in the section III we present a generic algorithm for DHF implementation and highlight the important steps. We describe different possible schemes that could be employed for each of those steps in the section IV. Section V starts with the description of the two models used in the study, followed by the subsection on the parameter settings and the simulation methodology. Results for proposed alternative methods are described in section VI, which is followed by the discussion in section VII. Finally the conclusion is given in section VIII.

II. HOMOTOPY BASED PARTICLE FLOW FILTERS

A. Bayesian recursive estimation

We start with the general formulation of bayesian recursive estimation for a markovian state space system.

Let $\mathbf{x}_k \in \mathbb{R}^d$ denote the state vector and $\mathbf{z}_k \in \mathbb{R}^m$ denote the measurement vector at time k . Also let \mathbf{Z}_k denote the set of measurements up to time k including \mathbf{z}_k , $\mathbf{Z}_k = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$. The state space model can be expressed in the terms of conditional probabilities,

$$\mathbf{x}_{k+1} \sim p(\mathbf{x}_{k+1} | \mathbf{x}_k) \quad (1)$$

$$\mathbf{z}_{k+1} \sim p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) \quad (2)$$

$p(\mathbf{x}_{k+1} | \mathbf{x}_k)$ and $p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})$ are referred to as the transition and the measurement/likelihood densities. Assuming additive process and measurement noises w_k and v_k we can write

$$p(\mathbf{x}_{k+1} | \mathbf{x}_k) = p_{w_k}(\mathbf{x}_{k+1} - \phi_k(\mathbf{x}_k)) \quad (3)$$

$$p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) = p_{v_k}(\mathbf{z}_{k+1} - \psi_k(\mathbf{x}_{k+1})) \quad (4)$$

where ϕ_k is termed as the process/dynamical model and ψ_k as the measurement model. According to the Chapman-Kolmogorov equation and the Bayes theorem, the prior density $p(\mathbf{x}_{k+1} | \mathbf{Z}_k)$ and the posterior density $p(\mathbf{x}_{k+1} | \mathbf{Z}_{k+1})$ are recursively defined as,

$$p(\mathbf{x}_{k+1} | \mathbf{Z}_k) = \int p(\mathbf{x}_{k+1} | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Z}_k) d\mathbf{x}_k \quad (5)$$

$$p(\mathbf{x}_{k+1} | \mathbf{Z}_{k+1}) = \frac{p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) p(\mathbf{x}_{k+1} | \mathbf{Z}_k)}{p(\mathbf{z}_{k+1} | \mathbf{Z}_k)} \quad (6)$$

where $p(\mathbf{x}_k | \mathbf{Z}_k)$ is posterior density at time k . These are also referred to as the process and measurement update equations respectively. The conditional density $p(\mathbf{z}_{k+1} | \mathbf{Z}_k)$ appears as a normalization constant in the measurement update formula, and it describes the distribution of measurement at time $k + 1$, conditioned on the set of all previous measurements. An exact closed form solution of (5) and (6) is generally not available for non-linear systems. Instead, two main approximate methods are used for the state estimation of such systems. In a first approach, the linearization of the model is performed around the current estimate (EKF) or the so called *Sigma-points* are propagated through the non-linear state space (UKF), thereby providing an approximation to the point estimates e.g. state mean and covariance. Another approach could be to numerically approximate the process and the measurement update equation. This could be done either by numerically evaluating the integrals over the discretized state space region [14], or by employing sequential Monte Carlo methods like particle filters [2], [3].

B. Derivation of generic homotopy flow equation

Log homotopy based particle flow filters also termed as the Daum-Huang particle flow filters (or simply the Daum-Huang filters DHF) as described in [9]–[15], share the importance sampling step with the SIR particle filter, but they specifically use the prior distribution of the state vector $p(\mathbf{x}_{k+1} | \mathbf{x}_k)$ as the importance density. The main difference lies in the way measurements are

incorporated to derive the posterior density. The idea here is to model the motion of particles from the prior to the posterior density, in a way analogous to the flow of physical particles. A log-homotopy function $\log p(\mathbf{x}_k, \lambda)$ is defined through the homotopy parameter λ ,

$$\log p(\mathbf{x}_{k+1}, \lambda) = \log g(\mathbf{x}_{k+1}) + \lambda \log h(\mathbf{x}_{k+1}) - \log K(\lambda). \quad (7)$$

where $g(\mathbf{x}_{k+1})$ represents the prior $p(\mathbf{x}_{k+1} | \mathbf{Z}_k)$, $h(\mathbf{x}_{k+1})$ the likelihood $p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})$ and λ the pseudo-time varying from 0 to 1. $K(\lambda)$ is the normalization constant independent of \mathbf{x}_{k+1} . $\lambda = 0$ sets $p(\mathbf{x}_{k+1}, \lambda)$ equal to the prior density while with $\lambda = 1$ the transformation is completed to the normalized posterior density. From now on we drop the time index k for the sake of convenience. It is supposed that the flow of particle obeys the Itô SDE,

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, \lambda)d\lambda + \boldsymbol{\sigma}(\mathbf{x}, \lambda)d\mathbf{w} \quad (8)$$

where $\mathbf{f}(\mathbf{x}, \lambda)$ is the flow vector, \mathbf{w} is the M-dimensional Wiener process with diffusion matrix $\boldsymbol{\sigma}(\mathbf{x}, \lambda)$. For a flow characterized as in (8), the evolution of the density $p(\mathbf{x}, \lambda)$ w.r.t. the parameter λ is given by the Fokker-Planck equation (also known as Kolmogorov forward equation),

$$\begin{aligned} \frac{\partial p(\mathbf{x}, \lambda)}{\partial \lambda} = & - \sum_{i=1}^d \frac{\partial}{\partial \mathbf{x}_i} [f_i(\mathbf{x}, \lambda)p(\mathbf{x}, \lambda)] \\ & + \frac{1}{2} \sum_{i=1}^d \sum_{j=1}^d \frac{\partial^2}{\partial \mathbf{x}_i \partial \mathbf{x}_j} [\mathbf{Q}_{i,j}(\mathbf{x}, \lambda)p(\mathbf{x}, \lambda)] \end{aligned} \quad (9)$$

where $\mathbf{Q}(\mathbf{x}, \lambda)$ is the diffusion tensor. This can be written in short hand notion,

$$\frac{\partial p(\mathbf{x}, \lambda)}{\partial \lambda} = -\nabla \cdot (\mathbf{f}(\mathbf{x}, \lambda)p(\mathbf{x}, \lambda)) + \frac{1}{2} \nabla^T \mathbf{Q}(\mathbf{x}, \lambda)p(\mathbf{x}, \lambda) \nabla \quad (10)$$

where ∇ is the spatial vector differentiation operator. From (7), the pseudo-time derivative of the density $p(\mathbf{x}, \lambda)$ can be formulated.

$$\frac{\partial p(\mathbf{x}, \lambda)}{\partial \lambda} = p(\mathbf{x}, \lambda) \left(\log h(\mathbf{x}) - \frac{\partial \log K(\lambda)}{\partial \lambda} \right) \quad (11)$$

By combining equations (10) and (11) we get,

$$\begin{aligned} p(\mathbf{x}, \lambda) \left(\log h(\mathbf{x}) - \frac{\partial \log K(\lambda)}{\partial \lambda} \right) \\ = -\nabla \cdot (\mathbf{f}(\mathbf{x}, \lambda)p(\mathbf{x}, \lambda)) + \frac{1}{2} \nabla^T \mathbf{Q}(\mathbf{x}, \lambda)p(\mathbf{x}, \lambda) \nabla \end{aligned} \quad (12)$$

Using the vector calculus identity,

$$\nabla \cdot (\mathbf{a}\mathbf{b}) = (\nabla \cdot \mathbf{a})\mathbf{b} + \mathbf{a} \cdot (\nabla \mathbf{b})$$

the equation 12 can be further expanded,

$$\begin{aligned} \log h(\mathbf{x}) - \frac{\partial \log K(\lambda)}{\partial \lambda} \\ = -\mathbf{f}^T(\mathbf{x}, \lambda) \cdot \nabla \log p(\mathbf{x}, \lambda) - \nabla \cdot \mathbf{f}(\mathbf{x}, \lambda) \\ + \frac{1}{2p(\mathbf{x}, \lambda)} (\nabla^T \mathbf{Q}(\mathbf{x}, \lambda)p(\mathbf{x}, \lambda) \nabla) \end{aligned} \quad (13)$$

The objective then becomes to solve the generic flow equation (13) for the yet unknown flow $\mathbf{f}(\mathbf{x}, \lambda)$.

C. Specific flow solutions

Various flow solutions have been obtained by solving (13) under different assumptions. Here we discuss four such flows derived by F. Daum and J. Huang in their series of papers.

1) Incompressible flow: The first solution of (13) appeared in [9], which was based on two distinct assumptions. Firstly, the diffusion term $\boldsymbol{\sigma}(\mathbf{x}, \lambda)$ in (8) is ignored. Secondly, the flow is considered incompressible, i.e. $\nabla \cdot \mathbf{f}(\mathbf{x}, \lambda) = 0$. Also the derivative of the log of normalization constant $\partial \log K(\lambda) / \partial \lambda$ is assumed to be very small, and therefore neglected. Applying these simplifications to the (13) leads to the incompressible flow equation,

$$\mathbf{f}^T(\mathbf{x}, \lambda) \cdot \nabla \log p(\mathbf{x}, \lambda) = -\log h(\mathbf{x}) \quad (14)$$

For one dimensional state space ($d=1$) the equation has an exact solution. However, for ($d \geq 2$) a simple inversion of the vector $\nabla(\log p(\mathbf{x}, \lambda))$ is not possible. Instead a unique minimum norm solution is obtained using the generalized inverse,

$$\mathbf{f}(\mathbf{x}, \lambda) = -\log h(\mathbf{x}) \frac{\nabla \log p(\mathbf{x}, \lambda)}{\|\nabla \log p(\mathbf{x}, \lambda)\|^2} \quad (15)$$

The authors suggest to use the fast kNN method to evaluate the gradients. Implementational details are described in [16]. Incompressible flow is generally inferior to exact flow [17]. Also it was reported in [18] that the incompressible flow could often hit a singularity for $d=1$. As a rebuttal to this, in [19] it has been argued that the incompressible flow can avoid singularities for $d \geq 2$, as singularities in higher dimension are just points in the state space and hence they can be bypassed/flown around. In the current work we will consider the incompressible flow for the sake of comparison. The filter based on the incompressible flow is termed as DHF-IC.

2) Exact flow: If the diffusion term is still assumed to be zero and $\partial \log K(\lambda) / \partial \lambda$ is neglected but the flow is allowed to be compressible, following equation can be derived from (13)

$$\log h(\mathbf{x}) + \mathbf{f}^T(\mathbf{x}, \lambda) \cdot \nabla \log p(\mathbf{x}, \lambda) = -\nabla \cdot \mathbf{f}(\mathbf{x}, \lambda) \quad (16)$$

Different flows have been derived in [20] based on solutions to the (16). One particular solution relates to

the case of $\log g(\mathbf{x})$ and $\log h(\mathbf{x})$ being bilinear in the components of vector \mathbf{x} , e.g., assuming a Gaussian prior and likelihood.

$$\log g(\mathbf{x}) = \log c_g - \frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{P}^{-1}(\mathbf{x} - \bar{\mathbf{x}}) \quad (17)$$

$$\log h(\mathbf{x}) = \log c_h - \frac{1}{2}(\mathbf{z} - \psi(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{z} - \psi(\mathbf{x})) \quad (18)$$

where $\log c_g$ and $\log c_h$ are the associated log normalization constants. The gradient of the two densities then can be written as,

$$\nabla \log p(\mathbf{x}, \lambda) = -\mathbf{P}^{-1}(\mathbf{x} - \bar{\mathbf{x}}) + \lambda \mathbf{H}^T \mathbf{R}^{-1}(\mathbf{z} - \psi(\mathbf{x})) \quad (19)$$

where $\mathbf{H} = (\partial\psi/\partial\mathbf{x})|_{\mathbf{x}_\lambda}$. More generally speaking, if the additive noise processes w_k and v_k belong to the exponential family, then an analytical solution termed as the *Exact flow* can be derived. For the Gaussian case, this is given as,

$$\mathbf{f}(\mathbf{x}, \lambda) = \mathbf{A}(\lambda)\mathbf{x} + \mathbf{b}(\lambda) \quad (20)$$

where $\mathbf{A}(\lambda)$ and $\mathbf{b}(\lambda)$ are,

$$\mathbf{A}(\lambda) = -\frac{1}{2}\mathbf{P}\mathbf{H}^T(\lambda\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H} \quad (21)$$

$$\mathbf{b}(\lambda) = (\mathbf{I} + 2\lambda\mathbf{A})[(\mathbf{I} + \lambda\mathbf{A})\mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{z} + \mathbf{A}\bar{\mathbf{x}}] \quad (22)$$

For nonlinear systems, the measurement model can be linearized by the Taylor series expansion up to the first term, such that $\mathbf{z} \approx \mathbf{z} - \psi(\mathbf{x}_\lambda) + \mathbf{H}\mathbf{x}_\lambda$. The derivation of the exact flow has been described in detail in [21]. We abbreviate this filter type as DHF-EF.

3) Coulomb's law based flow: Yet another solution can be developed in which the flow of particles in the pseudo-time is derived from the gradient of Poisson's equation [11]. Diffusion term in (13) is again assumed to be zero, but the derivative of the normalization constant is not ignored. Instead it is derived and an exact expression is found,

$$\frac{\partial \log K(\lambda)}{\partial \lambda} = \mathbb{E}(\log(h(\mathbf{x}))) \quad (23)$$

Then the equation (13) is written in the form

$$\nabla \cdot \mathbf{q}(\mathbf{x}, \lambda) = -\eta(\mathbf{x}, \lambda) \quad (24)$$

where $\mathbf{q}(\mathbf{x}, \lambda) = \mathbf{f}(\mathbf{x}, \lambda)p(\mathbf{x}, \lambda)$ and $\eta(\mathbf{x}, \lambda) = -p(\mathbf{x}, \lambda) \cdot (\log h(\mathbf{x}) - \partial \log K(\lambda)/\partial \lambda)$. It is noticed that the integral of $\eta(\mathbf{x}, \lambda)$ w.r.t. \mathbf{x} along the flow is zero,

$$\int_{\Omega} \eta(\mathbf{x}, \lambda) = 0 \quad (25)$$

where Ω is the relevant volume of the state space. This is analogous to the zero divergence of the electric flux density out of an enclosed region without any charge (first of the Maxwell's equations), i.e. net field lines entering the an enclosed region equal to the those leaving. Next it is reasoned that, if the function $\mathbf{q}(\mathbf{x}, \lambda)$ can be assumed to be the gradient of scalar potential

function $V(\mathbf{x}, \lambda)$, then the equation (24) can be expressed as the Poisson's equation for the potential $V(\mathbf{x}, \lambda)$.

$$\Delta V(\mathbf{x}, \lambda) = \eta(\mathbf{x}, \lambda) \quad (26)$$

such that,

$$f(\mathbf{x}, \lambda) = \frac{\nabla V(\mathbf{x}, \lambda)}{p(\mathbf{x}, \lambda)} \quad (27)$$

where Δ is the Laplacian operator. Solution to the (26) can be expressed in terms of the convolution integral for $d \geq 3$,

$$V(\mathbf{x}, \lambda) = - \int_{\Omega} \eta(\mathbf{y}, \lambda) \frac{c}{\|\mathbf{x} - \mathbf{y}\|^{d-2}} d\mathbf{y} \quad (28)$$

where $c = (4\pi)^{-d/2} \Gamma((d/2) - 1)$ and \mathbf{y} is the running variable. The above equation gives the solution of scalar potential $V(\mathbf{x}, \lambda)$, whereas our quantity of interest is its gradient. Taking the gradient of (28) we get,

$$\nabla V(\mathbf{x}, \lambda) = \mathbb{E} \left[(\log h(\mathbf{y}) - \mathbb{E}[\log h(\mathbf{x})]) \frac{c(2-d)(\mathbf{x} - \mathbf{y})^T}{\|\mathbf{x} - \mathbf{y}\|^d} \right] \quad (29)$$

Using (23) together with the Monte Carlo approximation for integrals, (29) can be approximated as,

$$\nabla V(\mathbf{x}_i, \lambda) \approx \frac{1}{k} \sum_{j \in S_i} \left(\log h(\mathbf{x}_j) - \frac{1}{k} \sum_{l \in S_i} \log h(\mathbf{x}_l) \right) \cdot \left(\frac{c(2-d)(\mathbf{x}_i - \mathbf{x}_j)^T}{\|\mathbf{x}_i - \mathbf{x}_j\|^d + \alpha} \right) \quad (30)$$

The expression for the gradient $\nabla V(\mathbf{x}, \lambda)$ is similar to the electromagnetic force equation given by Coulomb's law, hence the name of the flow. In order to reduce the computational complexity, the outer summation is carried over the subset of k nearest neighbors of the i th particle \mathbf{x}_i , which is denoted here by S_i . This is motivated by the fact that the as the state space dimension is increased, contribution of particles far apart, approaches zero exponentially. The inner expectation is approximated in a similar way. α is set to $(1/\beta)Tr(P)^{d/2}$, where both α and β are design parameters. Their purpose is to regularize the expression for $\nabla V(\mathbf{x}, \lambda)$. Also P can be approximated by a prior covariance matrix estimate. Filter based on this type of flow is referred as DHF-CLF

4) Non zero diffusion constrained flow: The last type of flow we consider, can be derived by not ignoring the diffusion term in equation (13) as suggested in [13]. Taking the gradient we get,

$$\begin{aligned} \nabla \log h(\mathbf{x}) &= -\nabla \log p(\mathbf{x}, \lambda)^T \cdot \nabla \mathbf{f}(\mathbf{x}, \lambda) \\ &\quad - \mathbf{f}^T(\mathbf{x}, \lambda) \cdot \nabla^2 \log p(\mathbf{x}, \lambda) - \nabla(\nabla \cdot \mathbf{f}(\mathbf{x}, \lambda)) \\ &\quad + \nabla \left(\frac{1}{2p(\mathbf{x}, \lambda)} \nabla^T \mathbf{Q}(\mathbf{x}, \lambda) p(\mathbf{x}, \lambda) \nabla \right) \end{aligned} \quad (31)$$

Analytical evaluation of the above equation for the flow $\mathbf{f}(\mathbf{x}, \lambda)$ is not possible, though numerical methods can be employed for this purpose. Depending on the dimensionality of the state-space, this could be computationally quite demanding. A little trick can lead to closed form solution for the flow, if the following constraint holds valid,

$$\begin{aligned} \nabla \left(\frac{1}{2p(\mathbf{x}, \lambda)} \nabla^T \mathbf{Q}(\mathbf{x}, \lambda) p(\mathbf{x}, \lambda) \nabla \right) \\ = \nabla \log p(\mathbf{x}, \lambda)^T \cdot \nabla \mathbf{f}(\mathbf{x}, \lambda) + \nabla (\nabla \cdot \mathbf{f}(\mathbf{x}, \lambda)) \end{aligned} \quad (32)$$

This results in a simple formula for the flow equation, given by

$$\mathbf{f}(\mathbf{x}, \lambda) = -(\nabla^2 \log p(\mathbf{x}, \lambda))^{-1} (\nabla \log h(\mathbf{x}))^T \quad (33)$$

The flow derivation does not involve neglecting the diffusion term, instead it appears in the constraint equation. Hence this flow is termed as non-zero diffusion constrained flow (NZDCF), and the DHF with this particular flow is termed as DHF-NZDCF.

A closer look at (33) reveals that it requires the Hessians of the log prior and the likelihood, as well as the gradient of the log-likelihood. The gradient and the Hessian of the log-likelihood, $\nabla \log h(\mathbf{x})$ and $\nabla^2 \log h(\mathbf{x})$, can be calculated analytically in most cases. On the other hand, there is no single method for the evaluation of the Hessian of the prior density, $\nabla^2 \log g(\mathbf{x})$. The most straight forward method is to approximate the prior density by a multivariate Gaussian density (e.g. using Laplace approximation), and use the negative of the inverse of the covariance matrix $-\mathbf{P}^{-1}$. This leads to the following,

$$\nabla^2 \log p(\mathbf{x}, \lambda) = \nabla^2 \log g(\mathbf{x}) + \lambda \nabla^2 \log h(\mathbf{x}) \quad (34)$$

$$\approx -\mathbf{P}^{-1} + \lambda \nabla^2 \log h(\mathbf{x}) \quad (35)$$

It has been suggested that the matrix \mathbf{P} can be set to the prior covariance matrix of a parallel running EKF/UKF. Another suggested method is to use the fast k-NN algorithm to compute the Hessian of the prior, similar to the incompressible flow. Alternatively, an approximation can be used instead, where \mathbf{P} is the state covariance matrix computed directly from the prior position of particles.

In this work, we primarily focus on analyzing DHF-NZDCF.

III. IMPLEMENTATION OF PARTICLE FLOW FILTERS

Numerical results for the DHF have been presented in [17]. DHF based on the incompressible and exact flows have been implemented by Choi. et.al. in [16] for non-linear scalar and linear vector system models. Exact flow DHF implementations for multi-target tracking using acoustic measurements have been reported in [21], where mobile targets are tracked based on their

received signal strength at a fixed receivers. In [22], joint probabilistic data association (JPDA) and maximum a posteriori penalty function (MAP-PF) algorithms based on the exact flow DHF have been derived. Recently, many researchers have carried out the comparative analysis for the DHF-NZDCF, in quite varied application. This include comparing the DHF performance against more traditional methods for angle only filtering in 3D by Gupta et.al. [23], comparing the tracking performance of DHF vs. other methods for supermaneuverable targets by Kreucher et.al. in [24], and the comparison of multisensor fusion using DHF against the particle filters by Mostagh and Chan in [25]. Results show a varying degree of success for DHF. While in some applications particle flow filters are shown to outperform the competitors, in others they do not perform quite well. The main issue is that while particle flow filters are theoretically quite elegant, their performance suffer from approximations made, both in theory and in the practical implementation. This include approximations made while deriving the flow, estimation of the prior density and the use of numerical techniques. This leads to the introduction of bias and loss of asymptotic consistency [26].

There could be several ways in which a DHF can be implemented. In Algorithm 1, we outline the method described by T. Ding and M. Coates in [21].

ALGORITHM 1: *Generic implementation of DHF*

```

Initialize DHF: Generate initial set of particles;
Initialize EKF/UKF: Initial mean and covariance;
Pseudo-time grid discretization;
for Loop over the time do
    Propagate particles using the dynamic model;
    Time update for EKF/UKF;
    Prior covariance matrix estimate from EKF/UKF;
    for Loop over the pseudo-time do
        for Loop over individual particles do
            | Integration of the flow equation;
            | end
        end
    Measurement Update for EKF/UKF;
    Redraw particles (Optional);
end

```

Particles are generated by sampling the transition density. An EKF/UKF is run in parallel to the main algorithm. This is done to approximate the prior covariance matrix. Next the flow equation is solved in the pseudo-time for all particles. The flow equation uses the prior covariance estimate from the parallel running EKF/UKF. Once done, the mean state vector is estimated and the measurement update is carried out for EKF/UKF. This process is repeated till the end of the simulation time. The steps colored in red are the crucial factors in the performance of the DHF.

The first is the pseudo-time λ discretization strategy together with the numerical integration method. As the DHF flow is described by an ordinary differential equation (ODE), a suitable discretization is essential to capture the flow dynamics. Then the flow equation is integrated w.r.t. λ . While the exact implementation details for references [13], [16],[22] are not clear, authors in [21] have used a single step Euler integration, as mentioned in the pseudo-code. It is simple to implement and is fairly quick. But care has to be taken as the flow ODE can exhibit stiffness. In that case a straight forward λ discretization together with the single step Euler integration might not work. Secondly, all flows described above require an estimate of the prior covariance matrix. While prior covariance estimate from parallel running EKF/UKF can be used as an approximation, this makes the DHF accuracy dependant on EKF/UKF. On the other hand, a sample covariance estimate can often be ill-conditioned. The question then becomes, is there a better method to estimate the prior covariance matrix. Finally, the re-generation of a new set of particles is an important step. Unlike a standard particle filter, the re-sampling/re-drawing step is not mandatory in the DHF, but optional. Instead, it has been mentioned that the homotopy flow moves the particles to their correct locations in the state space. But due to approximations made in the derivations, the flow may not be accurate, which could reduce the accuracy of the estimates. Hence the effect of particle re-generation is worth investigating. In the current work we look for improvements in the DHF performance by considering changes in the existing implementation architecture, as mentioned in Algorithm 1.

IV. IMPORTANT FACTORS IN DHF

In this section we individually discuss the aforementioned key factors affecting the DHF performance.

A. Pseudo-time discretization

While comparing the two flows, it was shown in [27] that the non zero diffusion flow is considerably stiffer as compared to the exact flow, where authors used 39 exponentially spaced λ points for solving the ODEs. This has also been mentioned in [13] where the usage of exponentially spaced time steps or higher order integration schemes is recommended to solve the issue. In this paper we consider both uniform and non-uniform grid discretization. The idea is to analyze the effect of a particular grid discretization strategy and the numerical integration scheme on the filter performance, in terms of the estimation error and the processing time. While the coarse λ discretization would not result in the correct solution, a fine discretization on the other hand would lead to a substantial increase in the computational cost. Therefore, a middle ground has to be chosen such that, flow dynamics at very small λ values are maximally captured, while only moderately increasing the processing cost.

B. ODE numerical solution

The homotopy flow is defined by a vector ordinary differential equation (ODE). In the current work, we seek for the numerical solution of the ODE. Broadly speaking, ODEs can be categorized into being stiff and non-stiff. While there is no precise definition of the stiffness, in the literature two criteria are generally mentioned for describing a stiff ODE. First, the condition number of the jacobian matrix $\mathbf{J}(x, \lambda) = \partial \mathbf{f}(x, \lambda) / \partial x$ of a stiff ODE is quite large. As a consequence, multiple timescales exist in the ODE. Time scales, often referred as modes, are defined by the inverse absolute eigenvalues of the Jacobian $J(x, \lambda)$. Secondly, in the Lipschitz's inequality $\|f(x_2, \lambda_2) - f(x_1, \lambda_1)\| \leq L \|\lambda_2 - \lambda_1\|$, the Lipschitz's constant L is typically very high for a stiff ODE. Non-zero diffusion ODE can be characterized as stiff according to both criteria. Therefore, care has to be taken when choosing the numerical integration scheme for solving the flow ODE.

The standard Euler's method has been used for solving the flow ODE in earlier works. It is a first order method with the truncation error in the order of $\mathcal{O}(h^2)$. In this paper, we intend to compare the performance of some other numerical integration (NI) schemes for solving stiff ODEs alongside the Euler's method. There are several choices available. Below, we mention some of the common NI methods for solving stiff ODEs.

1) Forth order Runge-Kutta method: Forth order Runge-Kutta method (RK4) is our second integration method. RK4 method has the local truncation error in the order of $\mathcal{O}(h^5)$, while the total accumulated error is of order $\mathcal{O}(h^4)$.

2) Rosenbrock method: Rosenbrock methods are family of multistep procedures to solve stiff ODEs. Jacobian matrix appears in the integration formula. Like the Runge-Kutta methods, Rosenbrock methods successively form intermediate results. If the Jacobian matrix is ignored then the method turns into the explicit Runge-Kutta scheme. Therefore, they are also called Runge-Kutta-Rosenbrock methods. Rosenbrock methods preserve exact conservation properties due to the use of the analytic Jacobian matrix, and possess optimal linear stability properties for stiff problems.

3) Gear's method: The Gear's method [28] belongs to the class of methods known as backward differentiation formulae (BDF). It is an implicit integration method and uses the first and higher order derivatives. Also, it is a predictor-corrector type scheme where each time step is initiated by prediction. Corrector iterations are then carried until prescribed convergence criteria are achieved or non-convergence is deemed to have occurred.

C. Prior covariance shrinkage estimation

The evaluation of the flow equation (33) require the availability of the prior covariance estimate. This

can be derived in several ways. The simplest way is to estimate the covariance matrix using the prior particles. This is referred to as the sample covariance estimate S . S is an unbiased estimator of the true prior covariance P , and is also the maximum likelihood estimate if the data is Gaussian distributed. But for non-linear models/non-Gaussian noises, the Gaussian assumption may not remain valid. Also S could progressively get ill-conditioned. i.e. the spread of the eigenvalues gets larger with the passage of time. This is especially the case, when the d/N_p ratio is non-negligible, where d is the state vector dimension and N_p is the number of particles. As a consequence, the matrix inversion could lead to stability issues. For the case of $d > N_p$, the resulting covariance matrix is not even full rank and hence not invertible. An alternative method suggested by authors in [15] is to run an EKF/UKF in parallel to DHF, and to use the prior covariance matrix generated by those filters. We refer to such matrix as P_{XKF} , where XKF could be Extended or Unscented version of the KF. While this method is better than using the raw data based covariance estimate, it ties the DHF estimation accuracy to that of the EKF/UKF. P_{XKF} could also exhibit a wide spread of the eigenvalues.

Therefore, we look for an alternative method for covariance matrix estimation. That method should have two properties: the resulting matrix should always be positive definite (PD) and the matrix should be well-conditioned [29]. One approach could be to start with the sample covariance, and ensure that the matrix is always PD. Such a matrix might not be well-conditioned. Alternatively, variance reduction techniques could be used to get a well-conditioned matrix, but this could be computationally expensive [30]. There is another approach used in the multivariate statistics literature for the estimation of the covariance matrices, known as the *shrinkage estimation*. The use of such methods dates back to work of Stein [31]. The main idea is to merge the raw estimate (S) which is unbiased but normally with high variance, together with a more structured but typically a biased target (B) through a scale factor, to get the combined estimate (P_*). The objective is to reduce the estimation error, typically in mean squared sense, by achieving an optimal trade off between the biased (B) and the unbiased (S) estimators. The scale factor is also called shrinkage intensity ρ as it shrinks the eigenvalues of S optimally towards the mean of eigenvalues of the true covariance matrix P [32]. The resulting covariance matrix (P_*) will be biased, but will improve on the two aforementioned properties, and is hoped to lower the estimation error. There are several shrinkage estimators mentioned in the literature, with different target covariance matrices. In the current work, we describe some of the more established shrinkage estimators. In subsections IV-C.1 to IV-C.3, shrinkage estimators are defined through a convex combination of the matrices B and

S . The objective becomes to find an optimal shrinkage intensity that minimizes the cost function,

$$\min_{\rho} E[\|P_* - P\|^2] \quad (36)$$

where $P_* = \rho B + (1 - \rho)S$.

1) Shrinkage towards the Identity matrix: Shrinkage towards the Identity matrix is described in [32]. The two main objectives defined are, to get an asymptotically consistent estimator that is more accurate than the sample covariance matrix S , and is also well-conditioned. No prior structure is assumed for the target matrix B , as it could lead to an increased biasness. Instead a simple matrix with same covariance terms and zero cross-variances (scaled Identity) is chosen as the target. The shrinkage estimator has following form

$$P_* = \frac{\alpha^2}{\delta^2} \mu^2 I + \frac{\beta^2}{\delta^2} S \quad (37)$$

The estimator P_* asymptotically shrinks its eigenvalues towards the mean eigenvalue of the true covariance matrix P , in quadratic mean sense. The terms α , β , δ and μ depend on the unobserved true covariance matrix P . Therefore, a consistent estimator of P_* is derived under the assumptions of general asymptotics. We term this estimator estimator is termed here as P_{LW0} , and has following form,

$$P_{LW0} = \frac{a_n^2}{d_n^2} m_n I + \frac{b_n^2}{d_n^2} S \quad (38)$$

where,

$$\begin{aligned} m_n &= \text{tr}(S)/d \\ d_n^2 &= \|S - m_n I\|^2 \end{aligned} \quad (39)$$

$$\text{xinonumber} \quad (40)$$

$$\bar{b}_n^2 = \frac{1}{n^2} \sum_{i=1}^{N_p} [\|x_i x_i^T - S\|^2]$$

$$b_n^2 = \min(\bar{b}_n^2, d_n^2)$$

$$a_n^2 = d_n^2 - b_n^2$$

$\|\cdot\|$ is the squared Frobenius norm and x_i is the i th particle. Also the shrinkage intensity ρ is given by a_n^2/d_n^2 . It is shown that the MSE for P_{LW0} asymptotically approaches that of P_* i.e. $\lim_{N_p \rightarrow \infty} E[\|P_* - P\|_{N_p}^2] - E[\|P_{LW0} - P\|_{N_p}^2] \rightarrow 0$. One main advantage of this estimator is that it does not assume any particular distribution for the data, and is therefore *distribution-free*.

2) Shrinkage towards the constant correlation matrix: This estimator is derived in [33], in the context of portfolio optimization. The target matrix is chosen according to the constant correlation model. It means that pairwise correlations are identical, which is given by the

average of all the sample correlations. We denote this estimator by P_{LW1} . The target matrix B is given by

$$B = \begin{cases} S_{ii} & i = j \\ \bar{r} \sqrt{S_{ii} S_{jj}} & i \neq j \end{cases}$$

where \bar{r} is the average sample correlation. It is defined as

$$\bar{r} = \frac{2}{d(d-1)} \sum_{i=1}^{d-1} \sum_{j=i+1}^d \frac{S_{ij}}{\sqrt{S_{ii} S_{jj}}} \quad (41)$$

The shrinkage intensity is defined as $\rho = \max\{0, \min\{1, \kappa/d\}\}$, with $\kappa = (\hat{\pi} - \hat{\varrho})/\hat{\gamma}$. $\hat{\pi}$ denotes the sum of asymptotic variances of the entries of the sample covariance matrix S , while $\hat{\varrho}$ denotes the sum of asymptotic covariances of the entries of the shrinkage target B with the entries of the sample covariance matrix. $\hat{\gamma}$ gives a measure of the misspecification of the shrinkage target. The hat ($\hat{\cdot}$) on the top of terms indicate the fact that these are the estimates of the true values, which are not known. $\hat{\pi}$ and $\hat{\varrho}$ are given by,

$$\begin{aligned} \hat{\pi} &= \frac{1}{d} \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^d \{(x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j) - S_{ij}\}^2 \\ \hat{\varrho} &= \sum_{i=1}^n \hat{\pi}_{ii} + \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\bar{r}}{2} \left(\sqrt{\frac{S_{jj}}{S_{ii}}} \vartheta_{ii,ij} + \sqrt{\frac{S_{ii}}{S_{jj}}} \vartheta_{jj,ij} \right) \end{aligned} \quad (42)$$

where,

$$\begin{aligned} \vartheta_{ii,ij} &= \frac{1}{d} \sum_{k=1}^d \{(x_{ik} - \bar{x}_i)^2 - \bar{x}_i\} \\ &\quad \cdot \{(x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j) - S_{ij}\} \\ \vartheta_{jj,ij} &= \frac{1}{d} \sum_{k=1}^d \{(x_{jk} - \bar{x}_j)^2 - \bar{x}_j\} \\ &\quad \cdot \{(x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j) - S_{ij}\} \end{aligned} \quad (43)$$

Finally $\hat{\gamma}$ is given by

$$\hat{\gamma} = \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n (B_{ij} - S_{ij})^2 \quad (44)$$

3) Shrinkage towards the perfect positive correlation matrix: Authors in [34] suggest single-factor matrix as the shrinkage target. The paper is concerned with estimating the structure of the risk in the stock market and the modelling of the stock returns. The fact that stock returns are positively correlated to each other, is exploited. The shrinkage target is given by,

$$B_{ij} = \begin{cases} S_{ii} & i = j \\ \sqrt{S_{ii} S_{jj}} & i \neq j \end{cases}$$

The resulting linear estimator is denoted as P_{LW2} . The shrinkage intensity has the same form as for P_{LW1} , but

with slightly different formula for $\hat{\varrho}$, which is given below.

$$\hat{\varrho} = \sum_{i=1}^n \hat{\pi}_{ii} + \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \frac{1}{2} \left(\sqrt{\frac{S_{jj}}{S_{ii}}} \vartheta_{ii,ij} + \sqrt{\frac{S_{ii}}{S_{jj}}} \vartheta_{jj,ij} \right) \quad (45)$$

4) Empirical bayesian: In [35], an estimator for multivariate gaussian data is derived. It is given by the linear combination of the sample covariance matrix S and scaled identity matrix. The scaling factor is estimated from the data. We denote this estimator by P_{EB} and it is given by

$$P_{EB} = \frac{N_p d - 2N_p - 2}{N_p^2 d} [\det(S)]^{1/d} I + \frac{N_p}{N_p + 1} S \quad (46)$$

5) Stein Haff: This estimator is described in [36]. The general form of the estimator is $V(S)\Phi(l(S))V(S)^T$, where $V(S)$ matrix contains the eigenvectors of the sample covariance matrix S while $\Phi(l(S))$ is a matrix that is a function of the eigenvalues $l(S)$ of the S . The data is assumed to be normally distributed, and the sample covariance estimate S is therefore Wishart distributed $S \sim \mathcal{W}(P, d)$. The Stein-Haff estimator denoted by P_{SH} , is constructed by leaving the eigenvectors of the S unchanged while replacing the eigenvalues l by $\tilde{l}_i = nl_i/(n - p + 1 + 2l_i \sum_{j=1, j \neq i}^{N_p} (1/(l_i - l_j)))$. Eigenvalues can get disordered by the transformation and might become negative, which could lead to the covariance estimate losing its positive definiteness. Therefore another algorithm called *isotonic regression* is used in conjunction with the transformation [37]. This lead to eigenvalues $\tilde{l} = \{\tilde{l}_1, \tilde{l}_2, \dots, \tilde{l}_p\}^T$. Hence, the estimate P_{SH} is given by $V(S)Diag(\tilde{l})V(S)^T$.

6) Minimax: The final shrinkage estimator considered is derived in [38]. Again Gaussian assumption is made. This estimator is termed minimax because under certain loss function, it has the lowest worst case error [32]. Its structure is similar to the P_{SH} but sample eigenvalues are replaced by $\tilde{l}_i = (n/(n + p - 1 - 2i)l_i)$. This estimator is denoted here by P_{MX} . Isotonizing regression is not applied in this case.

There is another interesting covariance estimator by Ledoit and Wolf [39] in which non-linear transformation of the sample eigenvalues is considered. Also it requires solving a non-linear optimization problem using sequential linear programming. It is shown that the new non-linear estimator outperforms the linear shrinkage estimators, described earlier in this section. In the current work we do not consider this method.

D. Re-generating the particles set

In the standard particle filter, new set of particles are generated after the measurement inclusion step. This is done in order to avoid the particle degeneration. A measure of the particle degeneracy is the effective number

of particles N_{eff} . When N_{eff} falls below a certain threshold, resampling of the particles is carried out. Depending on the number of particles, this can be computationally expensive. Homotopy based particle flow filters try to avoid the particle degeneracy by the gradual inclusion of the measurements. Unlike standard particle filters, resampling is not a mandatory step in the DHF [15], as it moves the particles to the correct region of the state-space. However due to the inexactness of the homotopy flow ODE, the particle state update itself is imperfect. Hence the generation of a new particle set could potentially help in relocating/confining the particles to the correct region. Instead of the conventional resampling, an optional redrawing of the particles is hinted out by the Daum and Huang in their papers. We find a single source describing the particles redrawing method. In [21], it is suggested to redraw a new set of posterior particles by sampling a Gaussian distribution. The mean of the distribution is estimated using particles, while the filtered covariance matrix is provided by the EKF. In the current work we consider two redrawing schemes, one using a single multivariate Gaussian distribution (MVG), and other using a Gaussian mixture model (GMM) that is estimated through the kernel density estimation (KDE).

1) MVG: Our first technique is inspired by the one described in the pseudo-code in [21]. The main difference is that we don't re-draw the whole set of particles. Instead, only those particles are redrawn which are deemed too *wayward*. Multivariate gaussian distribution is fitted to the posterior particles. This amounts to the estimation of the mean and variance of the MVG, given the particles. New particles are generated from this MVG.

2) KDE-GMM: Our next re-drawing scheme is based on the intuition that, a Gaussian fit to the posterior distribution might not be well suited for all cases. Hence we look for a non-Gaussian approximation to the filtered particles. The next most intuitive approach is to fit a Gaussian Mixture model (GMM) to the data. The key-factor in the GMM approximation is the number of components, which can be set to a fixed value or could be data driven. The textbook approach to estimate the GMM parameters is the expectation-maximization or EM method [40]. Alternatively, non-parametric methods like Kernel Density Estimation (KDE) may be employed for the estimation of the probability density, which is given by the sum of estimation kernels with a certain smoothing factor, centered at data points. Smoothing factor is also called bandwidth. In this paper we use the online KDE approach described by Kristan et al. in [41], in which a new method for online KDE is described. The method enables the construction of a multivariate probability density estimate by observing only a single sample at a time. The KDE of the target distribution is estimated using the sample distribution

which is constructed by online clustering of the data points. Each new observation is treated as a distribution in the form of Dirac delta functions. In the final form of the sample distribution, Dirac delta functions are smoothed out into Gaussians. Sample distribution is continuously refined and compressed in order to keep the algorithm complexity low.

3) Redrawing Algorithm: The purpose of re-drawing is to reduce the spread of the particles and relocate them in the appropriate region of the state space. We use the Mahalanobis distance (δ_M) for deciding the *waywardness* of particles. Given $\mathcal{N}(\mathbf{x} | \bar{\mathbf{x}}^p, P_p)$, the MVG approximation to the posterior, the distance (δ_M) for the posterior particle \mathbf{x}_i^p is given by,

$$\delta_M^{MVG}(i) = (\mathbf{x}_i^p - \bar{\mathbf{x}}^p)^T P_p^{-1} (\mathbf{x}_i^p - \bar{\mathbf{x}}^p) \quad (47)$$

We define a similar measure for the GMM model with K components,

$$\delta_M^{GMM}(i) = \sum_{k=1}^K w_k ((\mathbf{x}_i^p - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}_i^p - \mu_k)) \quad (48)$$

where w_k , μ_k and Σ_k are the weight, mean and covariance of the k th component of the GMM estimated through the online-KDE. Inverse of the distance $\varsigma = 1/\delta_M^*$ is a measure of the closeness of a particle to the estimated mean value. We use this value as a sort of weight ascribed to a particle, such that the particles close to the mean value are assigned a higher weight and vice versa. These weights are then normalized. Next, the particle *Assemblage*, denoted as Υ is calculated. Υ has the same form as the Effective Sample Size (ESS), in the traditional particle filter, and is a measure of the particle spread about the mean value. A higher value of Υ indicates an relatively even spread of the particles about the mean, whereas a lower value might suggest fragmentation of the particles into sub-clusters. A detailed analysis of this measure is presented in the Appendix A.

ALGORITHM 2: *Particle redrawing criterion*

```

 $\varsigma(i) = \frac{1}{\delta_M^*(i)} \forall i;$ 
 $\Upsilon = \frac{1}{\left( \frac{\sum_{i=1}^{N_p} \varsigma(i)}{\sum_{j=1}^{N_p} \varsigma(j)} \right)^2};$ 
if  $\Upsilon \leq \nu_M \cdot N_p$  then
  if  $\delta_M^*(i) \geq \sqrt{\frac{\Upsilon}{N_p}} \cdot \max \delta_M^* \forall i$  then
    | Redraw from  $\mathcal{N}(\mathbf{x} | \bar{\mathbf{x}}^p, P_p) / \sum_{j=1}^K w_k \mathcal{N}(\mathbf{x}_i | \mu_k, \Sigma_k);$ 
  else
    | NoRedraw;
  end
else
  | NoRedraw;
end

```

$$\begin{aligned}
\dot{x}_{k+1}^i &= \dot{x}_k^i + \dot{x}_k^i \Delta t + \frac{1}{2} a_{x_{k+1}}^i \Delta t^2 & \Pi_{x_k}^1 &= \frac{1}{N-1} \sum_{i=2}^N \left(\frac{\kappa_1}{\sqrt{(x_k^1 - x_k^i)^2 + (y_k^1 - y_k^i)^2 + \delta}} \right) \frac{v_r^2}{r_t} \cos\left(\frac{v_{tk}}{r_t}\right) \\
\dot{y}_{k+1}^i &= \dot{y}_k^i + \dot{y}_k^i \Delta t + \frac{1}{2} a_{y_{k+1}}^i \Delta t^2 & \Pi_{y_k}^1 &= -\frac{1}{N-1} \sum_{i=2}^N \left(\frac{\kappa_1}{\sqrt{(x_k^1 - x_k^i)^2 + (y_k^1 - y_k^i)^2 + \delta}} \right) \frac{v_t^2}{r_t} \sin\left(\frac{v_{tk}}{r_t}\right) \\
\dot{x}_{k+1}^i &= \dot{x}_k^i + \Pi_{x_k}^i \Delta t + a_{x_{k+1}}^i \Delta t & \Pi_{x_k}^i &= \kappa_2(x_k^1 - x_k^i) - \kappa_3 \dot{x}_k^i \\
\dot{y}_{k+1}^i &= \dot{y}_k^i + \Pi_{y_k}^i \Delta t + a_{y_{k+1}}^i \Delta t & \Pi_{y_k}^i &= \kappa_2(y_k^1 - y_k^i) - \kappa_3 \dot{y}_k^i \\
r_{k+1}^i &= \sqrt{(x_{k+1}^{(i)})^2 + (y_{k+1}^{(i)})^2} + v_{r_{k+1}}^i & \theta_{k+1}^i &= \tan^{-1}\left(\frac{y_{k+1}^{(i)}}{x_{k+1}^{(i)}}\right) + v_{\theta_{k+1}}^i
\end{aligned} \tag{D1}$$

$$\theta_{k+1}^i = \tan^{-1}\left(\frac{y_{k+1}^{(i)}}{x_{k+1}^{(i)}}\right) + v_{\theta_{k+1}}^i \tag{D2}$$

Gaussian noise noise:

$$\begin{aligned}
p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) &= p(\mathbf{r}_{k+1} | \mathbf{x}_{k+1}) p(\boldsymbol{\theta}_{k+1} | \mathbf{x}_{k+1}) \\
&= \frac{1}{(2\pi\sigma_r\sigma_\theta)^N} \prod_{i=1}^N \exp \left\{ -\frac{1}{2\sigma_r^2} \left(r_{k+1}^{(i)} - \sqrt{(x_{k+1}^{(i)})^2 + (y_{k+1}^{(i)})^2} \right)^2 - \frac{1}{2\sigma_\theta^2} \left(\theta_{k+1}^{(i)} - \tan^{-1}\left(\frac{y_{k+1}^{(i)}}{x_{k+1}^{(i)}}\right) \right)^2 \right\}
\end{aligned} \tag{D3}$$

Non-Gaussian noise:

$$\begin{aligned}
p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) &= p(\mathbf{r}_{k+1} | \mathbf{x}_{k+1}) p(\boldsymbol{\theta}_{k+1} | \mathbf{x}_{k+1}) \\
&= \frac{1}{(2\pi\beta^2)^{N/2} |\mathbf{R}_r|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{r}_{k+1} - \tilde{\mathbf{r}}_{k+1})^T \mathbf{R}_r^{-1} (\mathbf{r}_{k+1} - \tilde{\mathbf{r}}_{k+1}) \right\} \prod_{i=1}^N \exp \left\{ -\frac{1}{\beta} \left(\theta_{k+1}^{(i)} - \tan^{-1}\left(\frac{y_{k+1}^{(i)}}{x_{k+1}^{(i)}}\right) \right) \right\}
\end{aligned} \tag{D4}$$

$$\tilde{\mathbf{r}}_{k+1} = \left[\sqrt{(x_{k+1}^{(1)})^2 + (y_{k+1}^{(1)})^2} \sqrt{(x_{k+1}^{(2)})^2 + (y_{k+1}^{(2)})^2} \cdots \sqrt{(x_{k+1}^{(N)})^2 + (y_{k+1}^{(N)})^2} \right]^T \quad \mathbf{R}_r = \begin{bmatrix} \sigma_r^2 & \sigma_{r_x}^2 & \cdots & \sigma_{r_x}^2 \\ \sigma_{r_x}^2 & \sigma_r^2 & \cdots & \sigma_{r_x}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{r_x}^2 & \sigma_{r_x}^2 & \cdots & \sigma_r^2 \end{bmatrix}$$

Redrawing takes place only when the assemblage falls below a certain value, which in our case equals $\nu_M \cdot N_p$. We call ν_M as the *Redrawing Intensity* and its value can be set to any value between 0 and 1. When ν_M is 0, redrawing never takes place, while redrawing happens surely for the value 1. In our previous work [42], we re-drew the whole set of posterior particles, when the redrawing criterion was met. Here we make a small change and redraw only certain particles, which are deemed too off the mean value. For that purpose, we compare δ_M^* for each particle against a certain threshold which is dependant on the assemblage. If the criterion is met, the particle is redrawn from the MVG or GMM. The procedure is mentioned in the Algorithm 2.

V. MODEL DESCRIPTION

Here we consider a scenario similar to the one described in [27], namely the tracking of multiple targets

in a 2D space using range and bearing measurements, in order to study the effects of the methods proposed in the previous sections. States of targets are interdependent, therefore resulting in a non-linear coupled dynamical model. Furthermore, target association is assumed to be perfectly known and hence we do not use any data association algorithm. The state vector for the target i at time instant k is $\mathbf{x}_k^{(i)} = (x_k^{(i)}, y_k^{(i)}, \dot{x}_k^{(i)}, \dot{y}_k^{(i)})$, where $x_k^{(i)}$ and $y_k^{(i)}$ represent the position while $\dot{x}_k^{(i)}$ and $\dot{y}_k^{(i)}$ representing velocity components along the x and y-axis respectively. The overall state vector is formed by concatenating the individual target state vectors $\mathbf{x}_k = [\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)} \dots \mathbf{x}_k^{(N)}]$. Also the measurement vector for the target i is given by $\mathbf{z}_k^{(i)} = (r_k^{(i)}, \theta_k^{(i)})$, where $r_k^{(i)}$ is the range to the i th target while $\theta_k^{(i)}$ is its bearing. The overall measurement vector at time k is generated in a similar way. The process model is described in equations (D1), where $a_{x_{k+1}}^i$ and $a_{y_{k+1}}^i \sim \mathcal{N}(0, \sigma_a^2)$, Δt is the time discretization step size

and N is the total number of targets. The intuition behind the model is to make the targets motion coupled to each other. The target ($i = 1$) is pursued by all other targets ($i > 1$). The changes in the speed and direction of the targets depend on their relative distances to each other. κ_1, κ_2 and κ_3 are the coupling constants in the model. $\Pi_{x_k}^1$ and $\Pi_{y_k}^1$ control the speed/direction change for the pursued target and is inversely proportional to the sum of its relative distances to the all others. As pursuers come close, the pursued target changes its speed and direction. The direction change is realized through terms $(v_t^2/r_t) \cos((v_t/r_t)k)$ and $(v_t^2/r_t) \sin((v_t/r_t)k)$. r_t and v_t are the turning radius and velocity respectively and δ is a small offset. Similarly, the speed and direction changes for the pursuers are controlled by the terms $\Pi_{x_k}^i$ and $\Pi_{y_k}^i$.

If κ_1, κ_2 and κ_3 are set to zero, then state dynamics corresponds to the standard discrete white noise acceleration (DWNA) model. The measurement model for the i th target is given by (D2). Measurements consist of ranges and angles of the two object types, target and the pursuer. We consider two measurement models, one with uncorrelated Gaussian noises for both range and the angle, while the other with correlated Gaussian range noise and Exponentially distributed angle noise. For the first model, the likelihood is given by the (D3). We assume that both range and bearing measurement noise $v_{\theta_{k+1}}$ vectors are mutually independent at each time step. Also, both noises are uncorrelated within themselves such that $\mathbb{E}[v_{r_{k+1}}^i v_{r_{k+1}}^j] = 0$ and $\mathbb{E}[v_{\theta_{k+1}}^i v_{\theta_{k+1}}^j] = 0$ for $i \neq j$. In the second measurement model, range measurement noises $v_{r_{k+1}} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_r)$ are mutually correlated but are independent w.r.t. the bearing measurement noises $v_{\theta_{k+1}}$. Bearing measurement noise elements $v_{\theta_{k+1}}^i$ are exponentially distributed with the scale parameter β , such that $\mathbb{E}[(v_{\theta_{k+1}}^i)^2] = \beta^2$ and $\mathbb{E}[v_{\theta_{k+1}}^i v_{\theta_{k+1}}^j] = 0$. \mathbf{R}_r represent the covariance matrix of $v_{r_{k+1}}$ with $\sigma_r^2 = \mathbb{E}[(v_{r_{k+1}}^i)^2]$ and $\sigma_{r_x}^2 = \mathbb{E}[v_{r_{k+1}}^i v_{r_{k+1}}^j]$. $\sigma_{r_x}^2$ is assumed to be same for any two targets. Measurement noises are chosen as such in order to create a challenging estimation scenario, in which the relative strength of the particle flow method can be tested against the more traditional solutions like the EKF and the Particle filter. The likelihood function for this measurement model is given in (D4).

A. Parameters setting

We simulate two targets ($N=2$) in our analysis. Δt is set to 1, σ_a^2 to 0.5 ms^{-2} , σ_r^2 is set to 2000 m^2 , $\sigma_{r_x}^2$ to $(3/10)\sigma_r^2$, while β^2 is set to $(1/10)\text{rad}^2$. In this paper, we work only with the strongly coupled model with coupling constants κ_1, κ_2 and κ_3 set to 8000, 0.01 and 0.1 respectively. The turn radius r_t and turn speed v_t are set to 200 m and 10 ms^{-1} while δ is set to 0.001. We use 100 DHF particles ($N_p = 100$). DHF and SIR-PF particles are initialized by sampling Gaussian

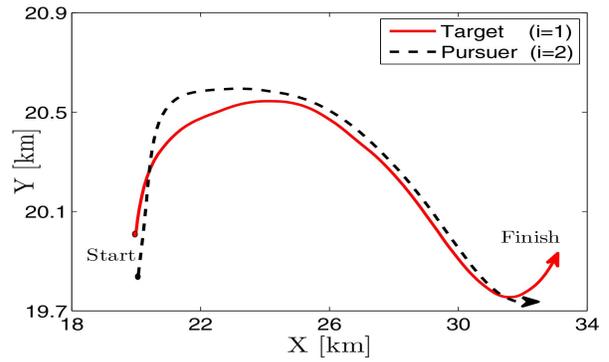


Fig. 1. Sample trajectory

distribution with mean of 20000 m and variance of 5000 m^2 for position elements, while their velocities are sampled from Gaussian distribution with mean and variance of 5 ms^{-1} and $25 \text{ m}^2 \text{ s}^{-2}$ respectively. EKF is initialized by sampling the Gaussian with initial state vector as mean and with variances 10^4 and 1 for the position and the velocity respectively. We note that $\sigma_r < D_{i,k} \sigma_\theta \forall i, k$, where $D_{i,k}$ represents the distance of i th target from the radar location at time instant k . In figure 1, we show a sample trajectory generated by using these parameters. We note that the target object ($i=1$) is pursued by the pursuing object ($i=2$). The target turns and increases speed as it is approached by the pursuer. The trajectory has segments of straight run as well as turns in the middle and at the end. Turning, in particular is challenging for the estimation algorithm, as this in addition to the non-linearity in the measurements, introduces non-linearity in the process model as well.

VI. SIMULATION RESULTS

We use root average mean square error (RAMSE) as the performance metric. It is defined as following. Let M be the total number of simulation runs for a particular scenario, $x_k^{i,m}$ and $y_k^{i,m}$ denote the positions of the i th target along X and Y-axis respectively, at time instant k in the m th trial. Likewise, let $\hat{x}_k^{i,m}$ and $\hat{y}_k^{i,m}$ denote estimated positions for the i th target. The RAMSE γ_r is then defined as,

$$\epsilon_r = \sqrt{\frac{1}{M} \sum_{m=1}^M \left[\frac{1}{N} \sum_{i=1}^N ((x_k^{i,m} - \hat{x}_k^{i,m})^2 + (y_k^{i,m} - \hat{y}_k^{i,m})^2) \right]}$$

We simulate each scenario for a total of fifty times ($M = 50$). First, we describe the effect of the numerical integration schemes.

A. Effect of numerical integration schemes

We compare the performance of the four methods mentioned in subsection IV-B, namely Euler's method, Runge-Kutta scheme of fourth order, Rosenbrock formula of second order and Gear's method. While we wrote scripts for the first two methods, MATLAB provided functions *ode23s* and *ode15s* were used for the

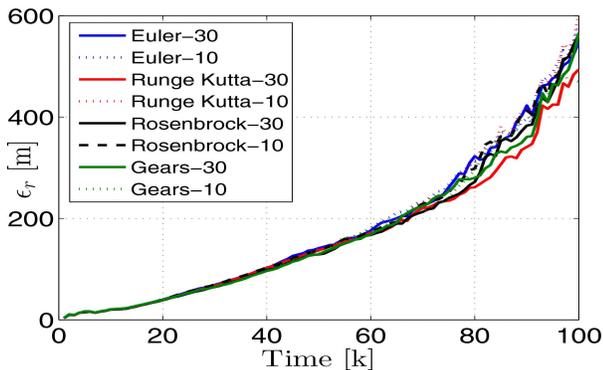


Fig. 2. Comparison of numerical integration schemes for the Gaussian noise

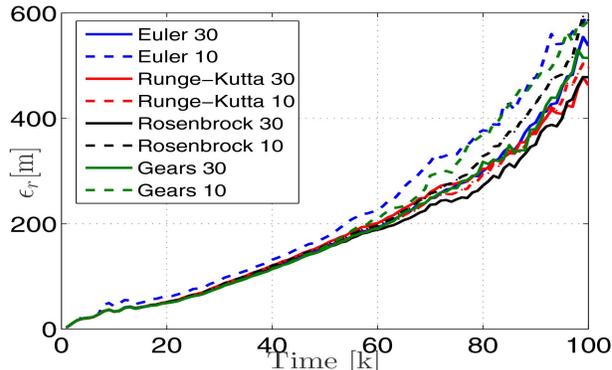


Fig. 3. Comparison of numerical integration schemes for the non-Gaussian noise

Rosenbrock and the Gear’s methods respectively. We also compare the effect of grid discretization on the performance of the above schemes. We use two specific cases, 10 uniformly spaced pseudo-time points (coarse discretization) and 30 exponentially spaced points (fine discretization). Also, the prior covariance matrix of a parallel running EKF is used to compute the flow. We plot the RAMSE ϵ_r for different schemes in figures 2a and 2b. We see a general increasing trend in the RAMSE vs. time for all methods. This is due to the specific process model used, which results in the peculiar targets trajectories involving rapid accelerations and sharp turns. It can be observed that the difference in the performance of different integration schemes is more pronounced in the case with non-Gaussian noise, as evident by the wider spread in the error curves. For the Gaussian case, we note that Runge-Kutta method with 30 λ points has the lowest error. Among the integration methods with 30 discretization points, Euler’s method has the highest error. We can also note that Gears-10 has the lowest error for all methods employing 10 discretization points. The largest error is exhibited by the Euler-10 method, which happens to be the fastest. On the other hand, Rosenbrock-30 is the slowest of all the methods. Euler-30 ranks second in the processing speed, as it is almost 1.5 times faster than its nearest competitor Runge-Kutta-10, while being 3 times as fast as Gears-10 though slightly inferior in the performance.

Next we discuss the results for the model with non-Gaussian measurement noise. As discussed earlier, the error curves show more spread. We note that the Rosenbrock method with 30 λ points has the lowest RAMSE, while the Euler’s scheme with 10 λ points is the worst performer followed closely by the Gears-10.

Runge-kutta methods with both 10 and 30 points are the second best. In fact, the difference in the performance between the two is very small. This is followed by the Gear-30 and the Euler-30 methods. We tabulate the time averaged RAMSE and the average processing time per particle for all methods in the Table I. Note that the time values mentioned only represent the time spent while solving the homotopy ODE for a single particle. The largest and the smallest values are highlighted in

TABLE I
Comparison for differ integration schemes

Gaussian		
Method	Avg. ϵ_r [m]	Proc.time (pp) [ms]
Euler-30	178.45	6.6
Euler-10	181.70	2.3
Runge-Kutta-30	163.06	27.4
Runge-Kutta-10	180.60	9.1
Rosenbrock-30	169.66	80.5
Rosenbrock-10	178.04	62.8
Gears-30	169.42	27.4
Gears-10	172.37	19.3
Non-Gaussian		
Method	Avg. ϵ_r [m]	Proc.time (pp) [ms]
Euler-30	186.69	5.6
Euler-10	223.07	1.8
Runge-Kutta-30	181.68	38.5
Runge-Kutta-10	184.39	12.7
Rosenbrock-30	173.17	71.9
Rosenbrock-10	196.30	55.8
Gears-30	184.49	26.4
Gears-10	186.69	17.9

red and green respectively. It can be seen that while the Runge-Kutta-30/Rosenbrock-30 are the best methods, they are also computationally very expensive. On the other hand, the Euler-10 is the fastest but the worst performer of all methods. Euler-30 represents a right trade-off between the performance and the processing time. In the proceeding analysis, we use Euler-30 as the default integration scheme.

B. Effect of shrinkage covariance estimation

Next we analyze the effect of shrinkage estimation schemes. We compare the performance of the six methods mentioned in subsection IV-C, together with that of sample covariance and the prior covariance matrices S and P_{EKF} respectively. We describe the DHF estimate generated using a particular covariance estimation scheme X as DHF- X . We use four metrics to judge the effectiveness of these methods. First and the foremost is the RAMSE of the DHF estimates. This is the central

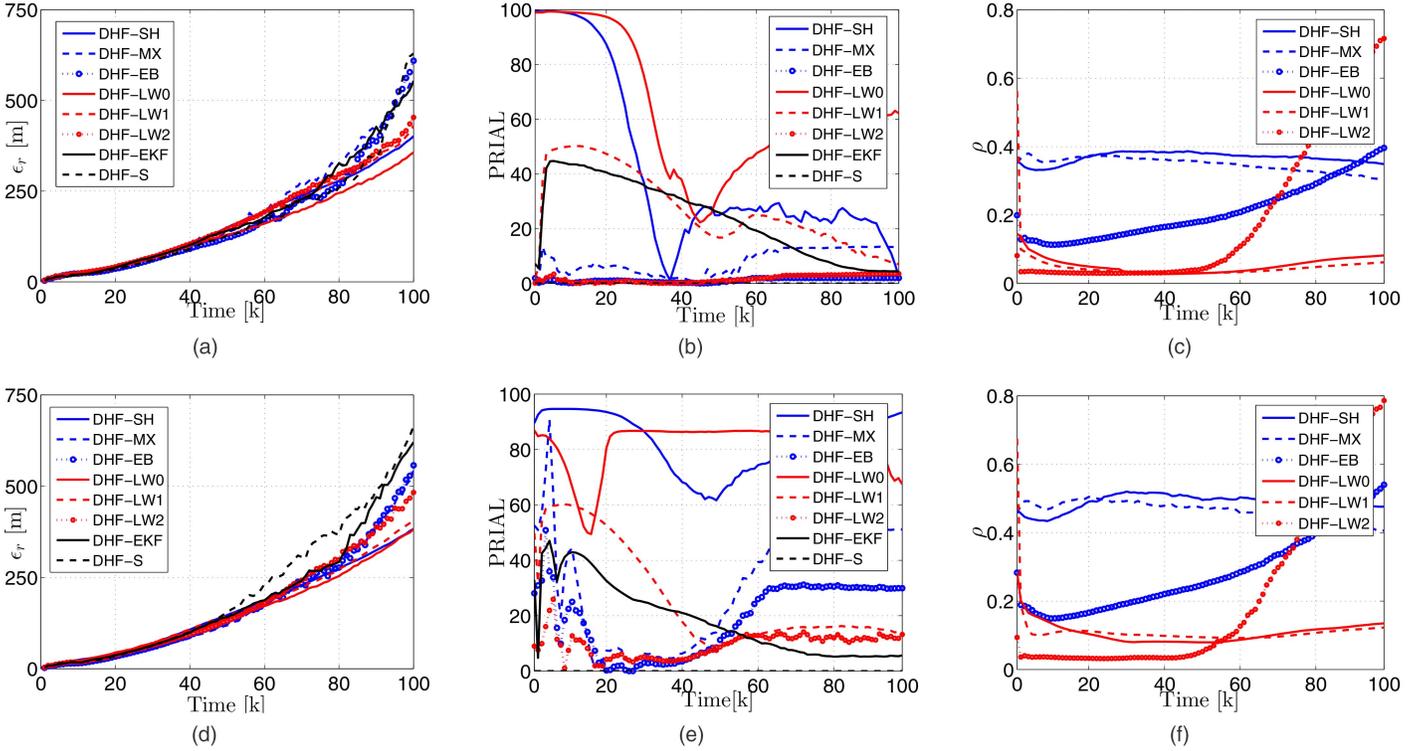


Fig. 4. (a, d) Position RAMSE (ϵ_r), (b, e) PRIAL and (c, f) Shrikage intensity (ρ) vs. time, for different covariance estimation schemes. Subfigures (a, b, c) show results for the case with Gaussian noise, while subfigures (d, e, f) show respective results for the case with non-Gaussian noise.

criterion for judging the effectiveness of the shrinkage schemes, in terms of the accuracy of the DHF estimates. Second is the relative accuracy of the covariance matrix estimates themselves. In the context of the shrinkage estimation, we use the percentage relative improvement in average loss or PRIAL as the measure for the exactness of any shrinkage covariance estimate, as defined in [32],

$$\text{PRIAL} = \left(1 - \frac{\mathbb{E}[\|P_{(\cdot)} - P\|^2]}{\mathbb{E}[\|S - P\|^2]} \right) \times 100 \quad (49)$$

where $\|(\cdot)\|$ represents the Frobenius norm, S is the sample covariance matrix estimate, while $P_{(\cdot)}$ and P are the shrunked covariance and the true covariance estimates, respectively. As P is not known, in the current scenario this is approximated by the covariance estimate from a sampling importance resampling particle filter (SIR-PF) with 25000 particles. Third, is the shrinkage intensity ρ , which indicates the compromise between the unbiased but more variant sample based estimate and the biased but less variant target. A lower value of ρ represents the closeness of covariance estimate to the Sample covariance matrix S . On the other hand, a higher value highlights a stronger influence of the target matrix B . At last, we use the condition number k_{cond} to analyze the spread in the eigenvalues of covariance estimates over the time. Plots for RAMSE, PRIAL and ρ are shown in figures 4 (a,d), 4 (b,e) and 4 (c,f) respectively,

while time averaged k_{cond} is shown only in the tabulated form in the Table II.

First we discuss the RAMSE for DHF with covariance estimates from all methods, for the Gaussian noise model. DHF-MX (Minimax) has the highest error. This can be explained as following. The minimax estimator scales the eigenvalues of the sample covariance matrix in a non-linear fashion. The highest $\lfloor ((p-1)/2) \rfloor$ eigenvectors have their eigenvalues shrunked, while for the others the eigenvalues are expanded. Scaling is done just based on the order of the sorted eigenvalues and it does not take into account any other possible information in the structure of the matrix S . This simplicity renders the estimator performing worse as compared to the others. Next in the line is the DHF-EKF. As can be seen in the figure 4d, the error increases sharply after about 80 s.

Although each simulated trajectory is not exactly the same, this is roughly the time when the targets start turning in our coupled motion model in most of those runs. Hence this is a critical point, as this tend to increase the non-linearity in our motion model. We see that for the DHF based on the EKF prior covariance, error starts rising indicating a failure in proper tracking. This outcome is inline with our previous results [27], where we reported that the standard DHF (EKF based DHF) fails for a coupled motion model. This also proves to be a very strong motivation for the search of an alternative covariance estimation method, which could be better than P_{EKF} . Interestingly, the performance of sample covariance based DHF is better than many other

TABLE II
Comparison for different covariance estimation schemes

Gaussian				
Method	Ave. ϵ_r [m]	Ave. PRIAL	Ave. ρ	Ave. k_{cond}
Stein-Haff	164.29	41.34	0.36	38620
Minimax	188.03	7.86	0.34	272820
Emp.Bayesian	170.94	1.20	0.20	181380
Ledoit-Wolf-0	144.77	63.13	0.05	170
Ledoit-Wolf-1	163.05	27.26	0.05	55610
Ledoit-Wolf-2	171.10	1.63	0.19	60370
EKF covariance	179.23	23.79	0	71460
Sample covariance	168.09	0	0	139760
Non-Gaussian				
Method	Ave. ϵ_r [m]	Ave. PRIAL	Ave. ρ	Ave. k_{cond}
Stein-Haff	161.22	83.30	0.40	45080
Minimax	166.58	32.0711	0.38	55490
Emp.Bayesian	171.28	18.78	0.23	46730
Ledoit-Wolf-0	153.32	81.71	0.09	180
Ledoit-Wolf-1	161.92	27.01	0.09	53220
Ledoit-Wolf-2	171.27	9.38	0.21	48470
EKF covariance	189.80	15.15	0	67770
Sample covariance	213.41	0	0	142260

schemes. In fact for most of the simulation time it has an error comparable to the better performing DHFs. It starts to increase only when targets start turning. After that time, the DHF-S fails to properly cope with the induced process non-linearity and the filter diverges rapidly. All variants of Ledoit-Wolf covariance estimators perform better, with LW0 based DHF outperforming all other filters. This can be attributed to the optimal convex combination (asymptotically) of the sample covariance matrix S and the scaled identity matrix I . This structure of the estimator results in a well-conditioned covariance estimator, that is more stable (from inversion point of view). This property can be critical when considering the turning motion of the targets, as DHF particles can be flung far and wide if the flow is incorrect which of course depends on inverting the prior covariance matrix. DHF with the other two covariance estimators from Ledoit and Wolf perform a little inferior relative to the DHF-LW0. P_{LW1} and P_{LW2} were derived for special problems in portfolio estimation and have very special structures. This lessens their generality and makes them very application specific.

Next we discuss the non-Gaussian case. We note that DHF-S is the worst method. DHF-EKF comes next as its error is also shows steeply diverging trend. This can be explained as follows: given that the measurements are non-linear functions of state variables, and bearing noise is exponentially distributed, the EKF is not a good approximation for the resulting non-linear and non-Gaussian scenario. Hence the covariance estimates generated by the EKF will not be accurate. DHF-LW0 has the lowest average error amongst all methods. This is because P_{LW0} is a distribution free estimator,

and hence produces good estimates even in this non-Gaussian scenario. It is followed by the Stein-Haff and Minimax estimators. Compared to the DHF-EKF, all estimators except the sample covariance DHF-S have lower average RAMSE.

Next we discuss the PRIAL for the covariance estimates. The expectation in the formula (49) is calculated by averaging over all simulation runs. A value of 100 means perfect estimation accuracy, while 0 means accuracy as good as the sample covariance matrix S . Again we discuss the Gaussian case first. We note that the PRIAL for P_{LW0} is highest while it is lowest for P_{LW2} . Again, this can be attributed to the very specific structure of this estimator. For the non-Gaussian case, we note that the PRIAL for P_{SH} is the highest on the average, while is lowest for the P_{LW2} .

One noteworthy thing is to compare the PRIAL of the estimators in the Gaussian vs. non-Gaussian case. We see that the PRIAL, on average, is lower for the Gaussian case. This can be explained by the fact that PRIAL represents how better an estimator when compared to the sample covariance estimator S . In non-Gaussian case, DHF-S is worse performer, which points to the fact that S is not a well-suited estimator. In fact, all DHFs are better than DHF-S. Hence we see that the PRIAL for the estimators in the non-Gaussian case is significantly higher. On the other hand in the case of Gaussian noise, S is not the worst estimator. This tends to increase the ratio $\mathbb{E}[\|P_c - P\|^2] / \mathbb{E}[\|S - P\|^2]$, which results in the lower values of PRIAL.

Shrinkage intensities ρ are shown in the figures 4c and 4f. We note that the lowest shrinkage intensity in both cases is exhibited by P_{LW0} . This suggest a higher contribution of the sample covariance than the scaled identity matrix in the optimal combination. P_{SH} has the highest shrinkage intensity on average and is also the most consistent. Shrinkage intensities in the non-Gaussian case are higher, again suggesting the inadequacy of the sample covariance matrix in the non-linear/non-Gaussian scenario. Finally we discuss the average logarithmic condition number $\log k_{cond}$. As expected, P_{LW0} has the lowest condition number over time, at least two orders of magnitude smaller than all other estimators. Also, the S has the highest condition number.

For the subsequent analysis, we consider P_{LW0} as the default covariance estimation scheme.

C. Effect of Redrawing

Once decided upon the pseudo-time discretization, flow integration and prior covariance estimation schemes, we now study the effect of redrawing on the performance of the DHF. As mentioned in section III-D, we consider two methods for regenerating the particles. The first method is redrawing from a Multivariate Gaussian (MVG), and the other from a Gaussian mixture model fitted to the posterior particles, that is estimated

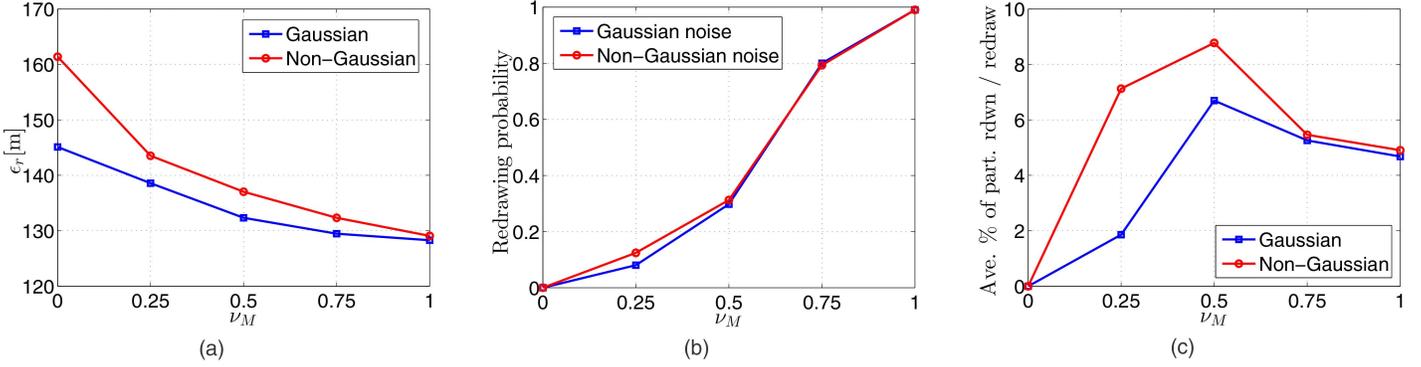


Fig. 5. (a) Time averaged RAMSE, (b) Redrawing probability and (c) Average percentage of particles redrawn vs. ν_M for Gaussian and non-Gaussian models.

using an online Kernel density estimation method. First we discuss the redrawing from MVG.

1) Multivariate Gaussian: We follow the Algorithm II mentioned in the section III-D for redrawing. The main parameter in that algorithm is the redrawing intensity ν_M . We vary ν_M between 0 and 1 and use five distinct values. First we study the effect of ν_M on the estimation accuracy, for which we plot the time averaged RAMSE for both Gaussian and the non-Gaussian cases in the Figure 5a. We see that as the redrawing threshold is increased the error decreases monotonically: the lowest error is for $\nu_M = 1$. We note that the improvement in the performance by increasing ν_M is stronger in non-Gaussian case than in the Gaussian one. This suggests the presence of more wayward particles in the non-Gaussian case, which are subsequently moved to the right regions after getting redrawn.

Next in the Figure 5b, we plot the redrawing probability vs. re-drawing intensity. Redrawing probability is defined as the number of times redrawing event takes place in the simulation divided by the total simulation time. So if particles are redrawn for half of the whole simulation duration, the redrawing probability is 0.5. The value is averaged over all the simulation runs. A higher value indicates a higher chance for particles to be redrawn during the simulation. We note a monotonically increasing relation between ν_M and the redrawing probability, which assumes a value of 1 for ν_M equals 1. This plot can also be used to infer about the assemblage, Υ . The assemblage is always greater than zero, making for the fact that no-redrawing happens for ν_M equals zero. As the ν_M is increased, the probability of finding Υ below the threshold $\nu_M \cdot N_p$ increases. e.g. from the figure 5b, it can be inferred that almost 30% of the time Υ value is below $0.5N_p$. This suggest that probability of having fragmentation of particles about the mean into two sub-groups of equal sizes (or any other equivalent scenario resulting in $\Upsilon=0.5$) is non-negligible. Also almost 50% of the time the value of the assemblage is between 50 and 75, while it is between 75 and 100 for almost 20% of the times. In relation to the RAMSE, we can conclude that the redrawing frequency has a direct

positive effect on the estimation error. A higher redrawing probability leads to the reduced estimation error. We note that both the Gaussian and non-Gaussian cases have a similar trend.

But how many particles, on average, are redrawn at a given time instance. While several metrics can be used for this effect, we use in particular the average percentage of particles redrawn, further averaged over the simulation time as plotted in the Figure 5c. We see an interesting trend. The percentage of particles redrawn increases with the increase in the intensity ν_M up to 0.5, at which it hits the maximum 7%–9% of the particles for both cases. Then this value decreases. This can be explained in the light of the redrawing probability. For ν_M between 0 and 0.5, the redrawing probability increases and so does the percentage of redrawn particles. This suggests that even though assemblage can be expected to be below $0.5N_p$ about 30% of the time, at times there is a significant number of particles satisfying the redrawing condition $\delta_M^*(i) \geq \sqrt{\Upsilon/N_p} \cdot \max \delta_M^*$. That is why the redrawing criteria $\Upsilon \leq \nu_M \cdot N_p$ is met in the first place, given the low value for ν_M . As ν_M is increased beyond 0.5, the redrawing probability increases, but the average number of particles satisfying the redrawing conditions decrease. That also points to the increase in the assemblage. We note that, on average, more particles are redrawn in the case of non-Gaussian noise than in the case of Gaussian case. This result is expected as estimation under the non-Gaussian noise is more challenging.

When seen together with the estimation error, we note that although the average rate of particles redrawn at any given time is not more than 10%, but redrawing those particles amounts to a significant reduction in the error. Also the particles redrawn for ν_M equals 1 have the maximum effect on the estimation error as they are the few quite separated from the rest of the particle cluster(s). If redrawn, they are moved to the correct region of the state-space, and hence contributing effectively to the point estimates.

2) Kernel Density Estimation: Now we discuss the effect of redrawing particles from a GMM, estimated

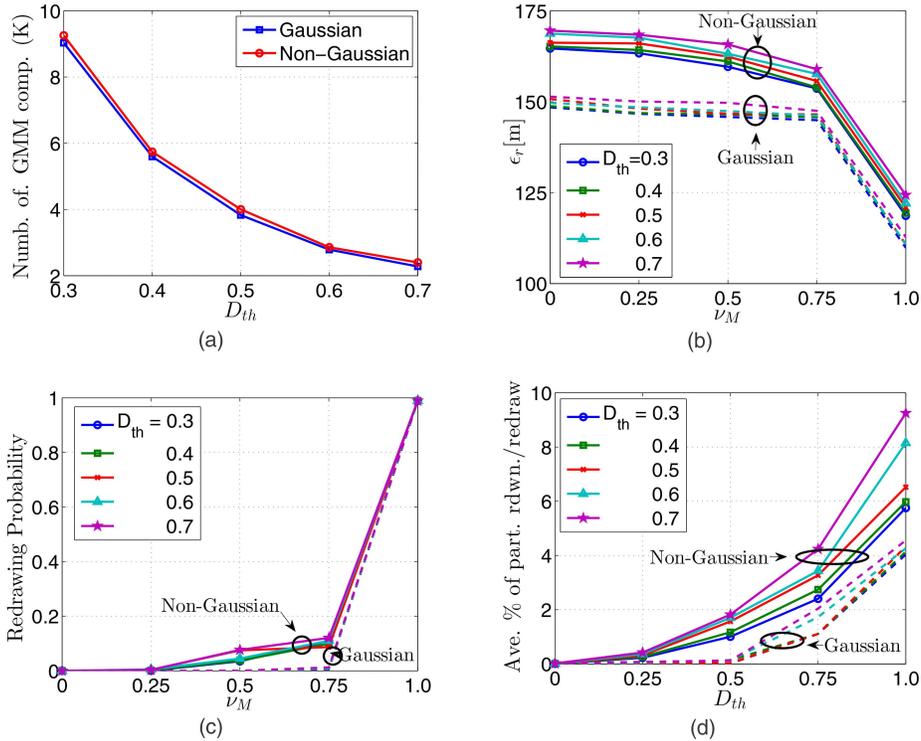


Fig. 6. (a) Average number of GMM components, (b) RAMSE vs. ν_M for different values of D_{th} , (c) Redrawing probability vs. ν_M and (d) Average number of particles redrawn per redraw vs. ν_M .

through the online KDE (oKDE) as described in [41], using the algorithm mentioned in the section III-D. We have used the source code for the oKDE provided by the authors at [43]. Although the method is general and can be used with any estimation kernel, the authors have used a multivariate Gaussian kernel in their work. oKDE method fits a GMM to the online data, which is supposed to arrive sequentially. In our context, we use the oKDE method to approximate the density of the particles after they move through the pseudo-time loop. Hence those particles can be thought of as coming from an importance sampler, and the task is to estimate the corrected posterior distribution. As a result we get an ensemble of weights, mean and covariances, $\{w_k, \mu_k, \Sigma_k\}_{k=1}^K$. Next, the averaged distance of each particle given the estimated GMM is calculated, and those particles which are thought to be too wayward are redrawn. As in the MVG case, we vary the redrawing threshold ν_M between 0 and 1.

There are two parameters that control the degree of estimation accuracy: the error threshold D_{th} , which controls the number of Gaussian components fitted to the data, and N_{init} which defines the number of data samples used for the initialization. Through experiments, we have found out that N_{init} , after a certain value, does not strongly influence the estimation accuracy. Therefore in our study we have kept N_{init} fixed to 33 (one third of total number of particles), while the threshold D_{th} is varied between 0.3 and 0.7, in the steps of 0.1. In the figure 6a we plot the average number of GMM

components (K) vs. the error threshold D_{th} . We note that as the D_{th} is increased, K decreases exponentially. This can be attributed to the particular implementation method used by the authors in [43].

Next in figure 6b, we show the results for position RAMSE vs ν_M for various values of threshold D_{th} , for both Gaussian and non-Gaussian cases. There are a number of noteworthy things. First, we note that the error for the Gaussian cases is less than that for the non-Gaussian, for all values of ν_M . We saw a similar behaviour in the previous section, where the redrawing was done using a MVG. Secondly, we see that the error only slightly decreases with increasing ν_M up to 0.75. After that we observe a significant reduction in the error for both cases. This is explained in the following way: in contrast to redrawing from a MVG where the particles far from the estimated mean value had lower weight defined by the ς , here such particles can be softly assigned to more than one Gaussian components. And due to the relative weights of the GMM components, the contribution of those particles is lessened. This results in a higher assemblage Υ value, and hence the redrawing criterion is rarely met. But when ν_M is sufficiently high, such that Υ is below $\nu_M \cdot N_p$, redrawing takes place. Particles which meet the redrawing condition are redrawn using the GMM. Statistically, particles are more likely to be redrawn from the components with the higher weights, and hence making those components even stronger while the opposite happens to the original low weight components. As a result, one can expect a

significant reduction in the particle spread after redrawn done in this manner. Lastly, we observe that the error for a lower value of D_{th} (hence higher K) is lower for both cases, for all values of ν_M . Again this is intuitive, as a higher number of GMM components is suggestive of the better accuracy of the fitted distribution to the posterior particles.

Figure 6c shows the redrawing probability vs. ν_M . We use the same definition for this probability as used in the previous section. We note that the redrawing probability for both noise cases is almost zero for ν_M less than or equal to 0.25. Between ν_M 0.25 and 0.75, we see a slight increase for the non-Gaussian case while it is still very close to zero for the Gaussian case. E.g. at $\nu_M = 0.75$, the redrawing probability is 10% for the case with non-Gaussian noise. A shape rise can be seen for both cases between 0.75 and 1. Also the redrawing probability is higher for the lower values of D_{th} . This trend has been explained in the previous paragraph, where it was mentioned that for the higher assemblage values, the probability of redrawing is quite low. Hence the redrawing probability also reveals information about the distribution of the assemblage. In contrast to the MVG case, the assemblage values are significantly larger (less spread). Therefore redrawing is only expected to happen for larger values of ν_M . Also a higher D_{th} (less K) tends to make the assemblage lower and hence the increasing the redrawing probability.

The average percentage of particles drawn per redraw is shown in the figure 6d. We observe a monotonically increasing trend for both Gaussian and non-Gaussian noises. We note that while the assemblage γ value effect the redrawing probability, it is the distribution of the Mahalanobis distance itself that influences the average percentage of particles drawn per redraw. From the results we can infer that Mahalanobis distance distributions for both Gaussian and non-Gaussian noises are similar, although for the latter it is more skewed towards the right, as evident from the higher percentage of redrawn particles. The average percentage of particles drawn per redraw rises sharply for ν_M between 0.75 and 1, hence more particles are redrawn for these values. This can be correlated with the large drop in the estimation error. Altogether, it can be inferred that the redrawing done for ν_M between 0.75 and 1 significantly increases the estimation accuracy. It can also be concluded that the GMM provides more accurate description for the posterior distribution. A higher percentage of particles is expected to be redrawn for higher values of D_{th} as the estimated GMM has fewer components, hence is not accurate enough.

D. Comparison against other filters

In this subsection, we compare the performance of our modified DHF against the other versions of DHF mentioned in the section II-C, together with the

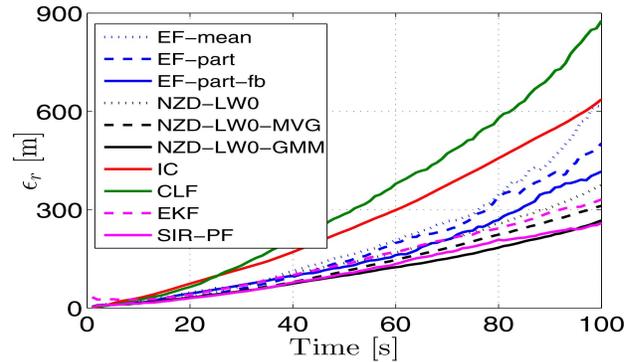


Fig. 7. Comparison for Gaussian noise

EKF and sampling importance resampling particle filter (SIR-PF) with 25000 particles. In total, we present the results for eight different variants for DHF, out of which three are different flavors of exact flow filter (EF), three are variants of non-zero diffusion constrained flow based filters (NZDCF), while the other two are based on the incompressible flow (IC) and the Coulomb's law flow DHF (CLF) respectively. The most basic version of the exact flow based DHF is reported in [16], where the flow equation is solved by linearizing the measurement model about the estimated prior mean value. We call this implementation as EF-mean. The second implementation of the exact flow has been reported by Ding and Coates in [21], and a pseudo-code is also provided. Two distinct changes are made to the EF-mean. In the first modification, the linearization of the measurement equation is carried out for individual particles, as opposed to being done only at the prior mean location. The second modification is related to the feedback of the DHF state estimates to the EKF, making the two filters coupled. In this study we consider these two cases individually i.e. the first modification alone and it together with the feedback. We call these implementations as EF-part and EF-part-fb respectively.

For the incompressible flow filter (IC), the flow equation (14) is solved for individual particles by assuming a Gaussian prior. Finally for the Coulomb's law based DHF (CLF), we use the parameters settings mentioned by the authors in [44]. One third of nearest neighbors are used in the evaluation of the equation (30). We have found that this filter is very sensitive to the parameters settings, and in general is very hard to tune. First we plot the RAMSE for the different filters for the Gaussian case in figure 7. We note that the CLF is the worst performer. The issue with this filter is the estimation of the probability density $p(\mathbf{x}, \lambda)$ for all particles throughout the pseudo-time, which is used in evaluating the flow equation $f(\mathbf{x}, \lambda) = \nabla \mathbf{V}(\mathbf{x}, \lambda) / p(\mathbf{x}, \lambda)$. As this is done using the few available particles, the resulting density estimate is not accurate enough and the filter is prone to divergence. This is also the issue with the Monte-Carlo approximation of the integral for gradient

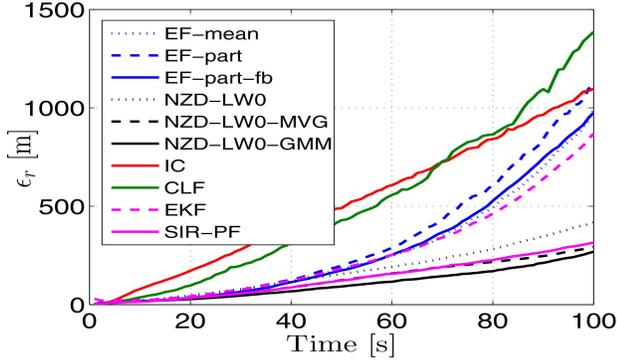


Fig. 8. Comparison for non-Gaussian noise

$\nabla V(\mathbf{x}, \lambda)$. Next we see that DHF based on IC although being better than with CLF, still fares worse compared to all other filters. The filter is based on the assumption of zero-divergence, which appears to be a quite strict condition. Also, the flow might encounter singularities which can make the filter diverge. Among the three variants of the exact flow, the EF-part-fb is the best. This is expected as for this filter linearization is done about each particle, and also the filter is coupled to a parallel running EKF. The best among all DHF variants are the ones based on NZDCF, all of which use Euler integration with 30 time steps and LW0 covariance estimation scheme. We denote the DHF-NZDCF without redrawing by NZD-LW0, with redrawing from MVG by NZD-LW0-MVG, and the one with redrawing from GMM as NZD-LW0-GMM. For the redrawing we set the threshold ν_M equal to 1. Also for the oKDE, we set the D_{th} equal to 0.5, which on average fits 4 GMM components to the posterior distribution. We note that the NZD-LW0-GMM is the best of all the schemes, even surpassing the SIR-PF with 25000. NZD-LW0-MVG is a little worse in performance to the SIR-PF, but is still better than the EKF. Next we discuss the results for the non-Gaussian measurement noise, as plotted in figure 8.

We note that all filters, except the variants of DHF-NZDCF and SIR-PF, perform poorly. DHF-IC and DHF-CLF fail to track the targets, with the latter being the worst in the performance. All variants of DHF-EF show a diverging error trend. This is due to the fact that EF hinges on the Gaussian assumption, which is not valid in the current case. As a part of the measurement noise is non-Gaussian, we see that these filters are unable to properly track targets. The same reasoning can be applied to EKF. NZD-LW0-GMM, NZD-LW0-MVG and SIR-PF are the first, second and the third best performer respectively. The error for all filters is generally larger when compared to the case with the Gaussian noise.

Next, we compare the execution time τ for a single update, including both the time and the measurement update steps. Matlab simulations were performed on the

TABLE III
Comparison of processing time for different filters

Method	Processing time τ [s] (Gaussian)	Processing time τ [s] (Non-Gaussian)
EKF	0.0004	0.0004
EF-mean	0.004	0.005
EF-part	0.10	0.10
EF-part-fb	0.105	0.105
IC	0.19	0.20
CLF	8.34	8.57
NZD	0.195	0.20
NZD-LW0	0.202	0.205
NZD-LW0-MVG	0.205	0.21
NZD-LW0-GMM ($D_{th}=0.3$)	1.77	1.83
NZD-LW0-GMM ($D_{th}=0.4$)	1.33	1.36
NZD-LW0-GMM ($D_{th}=0.5$)	1.19	1.21
NZD-LW0-GMM ($D_{th}=0.6$)	1.12	1.13
NZD-LW0-GMM ($D_{th}=0.7$)	1.09	1.11
SIR-PF ($N_p=25000$)	4.34	4.65

computer with Intel Core2 Quad with 2.66 GHz processors and 4 GB RAM. Table III shows the processing time per time step in seconds. We note that the EKF is the fastest of all methods. Next in the line are the EF based DHF, with DHF-EF-mean being the fastest. DHF with IC flow and NZD flow have quite similar processing time. We can also note that the covariance estimation (LW0) and redrawing from MVG do not incur any significant processing overhead. oKDE, on the other hand takes quite a while to compute the GMM components. The processing time is the highest for $D_{th}=0.3$ and it drops exponentially with increasing the threshold. Redrawing with threshold 0.5 takes almost 1.2 seconds per time step, which is 6 times the processing time of the DHF-NZD-LW0. Hence the redrawing with KDE takes significant amount of time. The particle filter with 25000 particles takes 4.5 seconds, which makes it almost 4 times slower than the DHF-NZD-LW0-GMM. Finally, the slowest method is the DHF-CLF taking almost 8.5 second per time step. We note that the processing time for the model with non-Gaussian noise is slightly higher in general for most of the schemes.

VII. DISCUSSION

Euler based numerical integration scheme is quite simple, but together with a clever pseudo-time discretization, can perform quite well. It is the most time efficient scheme. We analyzed different shrinkage estimation schemes. Some of them are tailor made for specific scenarios. The most general one is shrinkage towards identity matrix where no prior structure of the target matrix is assumed. It is a distribution free scheme and is shown to have outperformed other shrinkage estimators used in our analysis. Finally, we studied the effect of redrawing on the quality of the filter estimates. We choose two redrawing schemes: a

single MVG based re-drawing, and redrawing from a GMM estimated via the oKDE. The estimated density is then used to redraw particles which are considered too off the main cluster. The re-drawing algorithm uses the Mahalanobis distance of particles to calculate the assemblage \mathcal{Y} . When \mathcal{Y} falls below a certain threshold determined by the redrawing intensity ν_M , particles deemed too wayward are redrawn. We show that the redrawing, when combined with the shrinkage estimation reduces the error even further. Redrawing from a GMM gives better estimation accuracy than from the MVG.

VIII. CONCLUSION

DHF filters, even though not new in the literature, are still not fully explored in detail. They lack the in-depth theoretical and numerical analysis that the other contemporary filters have gone through. Especially, the implementational details are very application specific. In this paper we have tried to point out the key factors affecting the performance of a generic DHF. Highlighted factors have been studied individually in detail, with several possible methods suggested for each of them. This include different schemes for pseudo-time discretization, numerical integration, prior covariance estimation and the redrawing. We have compared their performance in a challenging non-linear multi-target scenario, under both Gaussian and non-Gaussian measurement noises. Eulers method with exponentially spaced pseudo-time points, provides a nice trade off between the performance and the complexity. DHF with shrinkage estimation methods is shown to have outperformed the one with the sample covariace matrix or with the EKF based estimate. Finally, it is shown that a NZDCF based DHF with the shrinkage estimation and proper redrawing, can outperform a bootstrap particle filter with comparable performance within less execution time.

ACKNOWLEDGMENT

We acknowledge the support by the EU's Seventh Framework Programme under grant agreement no 607400 (TRAX-Training network on tRacking in complex sensor systems) <http://www.trax.utwente.nl/>.

APPENDIX

A. Assemblage \mathcal{Y}

Let \mathbf{D} be the vector containing the Mahalanobis distances of the particles. We assume that the particles can be divided into L distinct sub-clusters, each cluster has the same distance to the estimated mean. In that case $\mathbf{D} = [d_1, d_2, \dots, d_L]$. This could either mean that the particles lie on hyper-balls in \mathbb{R}^d with radii d_i concentric around the estimated mean, or each cluster is small enough, and far apart from others, such that it can be approximated by individual hyper-balls. Let the i th

cluster has N_i number of particles such that $\sum_{i=1}^L N_i = N_p$.

Let the vector Φ contain the inverse of Mahalanobis distances

$$\Phi = \left[\left(\frac{1}{d_1} \right)_{\times N_1}, \left(\frac{1}{d_2} \right)_{\times N_2}, \dots, \left(\frac{1}{d_L} \right)_{\times N_L} \right]$$

The sum of the vector Φ is given by,

$$\begin{aligned} \sum_{i=1}^L \phi_i &= \frac{N_1}{d_1} + \frac{N_2}{d_2} + \dots + \frac{N_L}{d_L} \\ &= \frac{\sum_{i=1}^L \left(\prod_{j=1, j \neq i}^L d_j \right) N_i}{\prod_{j=1}^L d_j} \end{aligned}$$

Therefore the normalized vector $\tilde{\Phi}$ is given by

$$\begin{aligned} \tilde{\Phi} &= \left[\left(\frac{1}{d_1} \right)_{\times N_1}, \left(\frac{1}{d_2} \right)_{\times N_2}, \dots, \left(\frac{1}{d_L} \right)_{\times N_L} \right] \\ &\quad \times \frac{\prod_{j=1}^L d_j}{\sum_{i=1}^L \left(\prod_{j=1, j \neq i}^L d_j \right) N_i} \end{aligned}$$

Next the sum of squares of the above vector is evaluated,

$$\begin{aligned} \sum_{i=1}^L \tilde{\phi}_i^2 &= \left[\frac{N_1}{(d_1)^2} + \frac{N_2}{(d_2)^2} + \dots + \frac{N_L}{(d_L)^2} \right] \\ &\quad \times \left(\frac{\prod_{j=1}^L d_j}{\sum_{i=1}^L \left(\prod_{j=1, j \neq i}^L d_j \right) N_i} \right)^2 \\ &= \frac{N_1 \prod_{j=1, j \neq 1}^L (d_j)^2}{\sum_{i=1}^L \left(\prod_{j=1, j \neq i}^L (d_j)^2 \right) N_i} \dots \\ &\quad \dots + \frac{N_2 \left(\prod_{j=1, j \neq 2}^L d_j \right)^2}{\left(\sum_{i=1}^L \left(\prod_{j=1, j \neq i}^L d_j \right) N_i \right)^2} + \dots \\ &\quad \dots + \frac{N_L \left(\prod_{j=1, j \neq L}^L d_j \right)^2}{\left(\sum_{i=1}^L \left(\prod_{j=1, j \neq i}^L d_j \right) N_i \right)^2} \end{aligned}$$

$$\sum_{i=1}^L \tilde{\phi}_i^2 = \sum_{i=1}^L N_i \left(\frac{\prod_{j=1, j \neq i}^L d_j}{\sum_{k=1}^L \left(\prod_{j=1, j \neq k}^L d_j \right) N_k} \right)^2$$

and finally the assemblage \mathcal{Y} is given by,

$$\mathcal{Y} = \frac{1}{\sum_{i=1}^L \tilde{\phi}_i^2} = \frac{\left(\sum_{k=1}^L N_k \left(\prod_{j=1, j \neq k}^L d_j \right) \right)^2}{\sum_{i=1}^L N_i \left(\prod_{j=1, j \neq i}^L d_j \right)^2}$$

Below, we consider few special cases for the assemblage.

1) Number of clusters equals N_p : Each particle is considered as single clusters, hence each clusters has one particle with distinct distance d_i . assemblage in that case is given by,

$$\Upsilon = \frac{\left(\sum_{k=1}^{N_p} \left(\prod_{j=1, j \neq k}^{N_p} d_j\right)\right)^2}{\sum_{i=1}^{N_p} \left(\prod_{j=1, j \neq i}^{N_p} d_j\right)^2}$$

2) All particles equidistant: If $d_i \approx d$

$$\Upsilon = \frac{\left(\sum_{k=1}^{N_p} d^{N_p-1}\right)^2}{\sum_{k=1}^{N_p} (d^{N_p-1})^2} = \frac{(N_p d^{N_p-1})^2}{N_p d^{2(N_p-1)}} = \frac{N_p^2 d^{2(N_p-1)}}{N_p d^{2(N_p-1)}}$$

which leads to,

$$\Upsilon = N_p$$

3) Two dominant clusters: Now suppose that there are two main sub-cluster i.e. $L=2$.

$$\Upsilon = \frac{\left(\sum_{k=1}^2 N_k \left(\prod_{j=1, j \neq k}^2 d_j\right)\right)^2}{\sum_{i=1}^2 N_i \left(\prod_{j=1, j \neq i}^2 d_j\right)^2} = \frac{(d_2 N_1 + d_1 N_2)^2}{d_2^2 N_1 + d_1^2 N_2}$$

Now assume that $d_1 \gg d_2$. In that case we can say in the limiting sense,

$$\begin{aligned} \lim_{d_1 \rightarrow \infty} \Upsilon &= \lim_{d_1 \rightarrow \infty} \frac{(d_2 N_1 + d_1 N_2)^2}{d_2^2 N_1 + d_1^2 N_2} \\ &= \frac{N_2^2}{N_2} = N_2 \end{aligned}$$

Likewise for $d_2 \gg d_1$, $\lim_{d_2 \rightarrow \infty} \Upsilon = N_1$.

REFERENCES

- [1] F. Gustafsson
Statistical sensor fusion, 2nd ed.
Studentlitteratur AB, May 21, 2013.
- [2] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp
“A tutorial on Particle Filters for Online Non-linear/Non-gaussian Bayesian Tracking,”
in *IEEE transaction on signal processing*, vol. 50, no. 2. IEEE, February 2002, pp. 174–188.
- [3] N. Gordon, D. Salmond, and A. Smith
“Novel approach to nonlinear/non-Gaussian bayesian state estimation,”
in *IEE Proceedings on Radar and Signal Processing.*, Apr 1993, pp. 107–113.
- [4] M. Pitt and N. Shephard
“Filtering via Simulation: Auxiliary Particle Filters,”
in *Journal of the American Statistical Association*, vol. 94, no. 446, 1999.
- [5] C. Musso, N. Oudjane, and F. Le Gland
“Improving Regularised Particle Filters,”
in *Sequential Monte Carlo Methods in Practice*, ser. Statistics for Engineering and Information Science, A. Doucet, N. de Freitas, and N. Gordon, Eds. Springer New York, 2001, pp. 247–271.
- [6] U. Hanebeck and O. Feiermann
“Progressive Bayesian estimation for nonlinear discrete-time systems: the filter step for scalar measurements and multidimensional states,”
in *IEEE Conference on Decision and Control*, vol. 5. IEEE, 9–12 Dec. 2003, pp. 5366–5371.
- [7] J. Hagmar, M. Jirstrand, L. Svensson, and M. Morelande
“Optimal Parameterization of Posterior Densities Using Homotopy,”
in *Proceedings of the 14th International Conference on Information Fusion (FUSION)*. 5–8 July: IEEE, 2011, pp. 1–8.
- [8] F. Daum and J. Huang
“Curse of Dimensionality and Particle Filters,”
in *IEEE Aerospace conference*. IEEE, 8–15 March 2003, pp. 1970–1993.
- [9] F. Daum and J. Huang
“Nonlinear filters with log-homotopy,”
in *SPIE Proceedings*, September 2007.
- [10] F. Daum and J. Huang
“Nonlinear filters with particle flow induced by log-homotopy,”
in *SPIE Proceedings*, May 2009.
- [11] F. Daum, J. Huang, and A. Noushin
“Coulomb’s law particle flow for nonlinear filters,”
vol. 8137, 2011, pp. 81 370E–81 370E-15.
- [12] F. Daum and J. Huang
“Zero curvature particle flow for nonlinear filters,”
in *SPIE Proceedings*, 2013.
- [13] F. Daum and J. Huang
“Particle flow with non-zero diffusion for non-linear filters,”
in *SPIE Proceedings*, 2013.
- [14] M. A. Khan, M. Ulmke, B. Demissie, F. Govaers, and W. Koch
“Combining Log-Homotopy Flow with Tensor decomposition based solution for Fokker-Planck equation,”
in *2016 19th International Conference on Information Fusion (FUSION)*, July 2016, pp. 2229–2236.
- [15] F. Daum and J. Huang
“Nonlinear filters with particle flow,”
in *SPIE Proceedings*, September 2009.
- [16] S. Choi, P. Willet, F. Daum, and J. Huang
“Discussion and Application of Homotopy filter,”
in *SPIE Proceedings*, 2011.
- [17] F. Daum and J. Huang
“Numerical experiments on nonlinear filters with exact particle flow induced by log-homotopy,”
in *Proceeding SPIE*, April 2010.
- [18] L. Chen and R. Mehra
“A study of nonlinear filters with particle flow induced by log-homotopy,”
in *SPIE Proceedings*, 2010.
- [19] F. Daum and J. Huang
“Friendly rebuttal to Chen and Mehra: incompressible particle flow for nonlinear filters,”
vol. 8392, 2012, pp. 839 210–839 210-6. [Online]. Available: <http://dx.doi.org/10.1117/12.915510>.
- [20] F. Daum and J. Huang
“Generalized particle flow for nonlinear filters,”
in *SPIE Proceeding*, April 2010.
- [21] T. Ding and M. Coates
“Implementation of Daum-Huang Exact Flow Particle Filter,”
in *IEEE Statistical Signal Processing Workshop (SSP)*, 2012.
- [22] K. Bell and L. Stone
“Implementation of the homotopy particle filter in the JPDA and MAP-PF multi-target tracking algorithms,”
in *Information Fusion (FUSION), 2014 17th International Conference on*, July 2014, pp. 1–8.
- [23] S. Datta Gupta, J. Y. Yu, M. Mallick, M. Coates, and M. Morelande
“Comparison of angle-only filtering algorithms in 3d using EKF, UKF, PF, PFF, and ensemble KF,”
in *Information Fusion (Fusion), 2015 18th International Conference on*, July 2015, pp. 1649–1656.

- [24] C. Kreucher, K. Bell, and D. Sobota
 “A comparison of tracking algorithms for supermaneuverable targets,”
 in *Information Fusion (Fusion), 2015 18th International Conference on*, July 2015, pp. 534–541.
- [25] N. Moshtagh and M. Chan
 “Multisensor fusion using homotopy particle filter,”
 in *Information Fusion (Fusion), 2015 18th International Conference on*, July 2015, pp. 1641–1648.
- [26] P. Bunch and S. Godsill
 “The Progressive Proposal Particle Filter: Better Approximations to the Optimal Importance Density,”
 Tech. Rep., 2014.
- [27] M. Khan and M. Ulmke
 “Non-linear and non-Gaussian state estimation using log-homotopy based Particle Flow Filters,”
 in *Sensor Data Fusion: Trends, Solutions, Applications (SDF), 2014*, Oct 2014, pp. 1–6.
- [28] C. W. Gear
 “The automatic integration of stiff ordinary differential equations,”
 in *Information processing 68 (Proc. IFIP Congress, Edinburgh, 1968)*, vol. 1, 1969, pp. 187–193.
- [29] J. Schäfer and J. Strimmer
 “A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics,”
 2005.
- [30] J. Schäfer and J. Strimmer
 “An empirical Bayes approach to inferring large-scale gene association networks,”
Bioinformatics, vol. 21, pp. 754–764, 2005.
- [31] C. Stein
 “Inadmissibility of the usual estimator for the mean of a multivariate normal distribution,”
 in *Proceedings of the Third Berkeley Symposium on Mathematical and Statistical Probability.*, vol. 1, no. 446, 1956, pp. 197–206.
- [32] O. Ledoit and M. Wolf
 “Well-conditioned estimator for large-dimensional covariance matrices,”
Journal of Multivariate Analysis, vol. 88, no. 2, pp. 365–411, 2004.
- [33] O. Ledoit and M. Wolf
 “Honey, I shrunk the sample covariance matrix,”
The Journal of Portfolio Management, vol. 30, no. 4, pp. 110–119, 2004.
- [34] O. Ledoit and M. Wolf
 “Improved estimation of the covariance matrix of stock returns with an application to portfolio selection,”
Journal of Empirical Finance, vol. 10, no. 5, pp. 603–621, 2003.
- [35] L. Haff
 “Empirical Bayes Estimation of the Multivariate Normal Covariance Matrix,”
Ann. Statist., vol. 8, no. 3, pp. 586–597, 1980.
- [36] C. Stein
 “Estimation of a covariance matrix, Rietz Lecture,”
 39th Annual Meeting IMS, Atlanta, GA, Tech. Rep., 1975.
- [37] S. Lin and M. Perlman
 “A Monte Carlo Comparison of four estimators of a Covariance Matrix,”
 Tech. Rep. 44, 1984.
- [38] C. Stein
 “Series of lectures given at the University of Washington,”
 Seattle, Tech. Rep., 1982.
- [39] O. Ledoit and M. Wolf
 “Nonlinear shrinkage estimation of large-dimensional covariance matrices,”
The Annals of Statistics, vol. 40, no. 2, pp. 1024–1060, April 2012.
- [40] C. M. Bishop
Pattern Recognition and Machine Learning.
 Springer, 2006.
- [41] M. Kristan, A. Leonardis, and D. Skočaj
 “Multivariate Online Kernel Density Estimation with Gaussian Kernels,”
Pattern Recognition, vol. 44, pp. 2630–2642, 2011.
- [42] M. A. Khan and M. Ulmke
 “Improvements in the Implementation of Log-Homotopy based Particle Flow Filters,”
 in *Information Fusion (Fusion), 2015 18th International Conference on*, July 2015, pp. 74–81.
- [43] “Online Kernel Density Estimation with Gaussian Kernels and Online Discriminative Kernel Density Estimation with Gaussian Kernels,”
http://www.vicos.si/File:Maggot_v3.4.zip.
- [44] F. Daum, J. Huang, and A. Noushin
 “Numerical experiments for Coulomb’s law particle flow for nonlinear filters,”
 vol. 8137, 2011.



Muhammad Altamash Ahmed Khan received his B.E. in electrical engineering from the National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2006 and the M.Sc. in wireless systems from KTH Royal Institute of Technology, Stockholm, Sweden in 2011. From 2011 to 2013, he worked as Research Engineer with the Lab for Automatic Control and at the Lab for Communication Networks, KTH. During that period he worked on the statistical characterization of image/video feature extraction, implementation and testing of motion control for networked robots, and implementing the routing and MAC protocols for wireless sensor networks. Since April 2014, he has been working as a Marie Curie Fellow at FKIE Fraunhofer, Wachtberg Germany. His research interests include non-linear state estimation methods, multi-target tracking, and wireless sensors networks.



Martin Ulmke is a theoretical physicist and received his Dipl.-Phy. (M.S.) and Dr. rer. nat. (Ph.D.) degrees from the Aachen Technical University (RWTH), Germany, in 1991 and 1995, respectively. From 1995 to 1998 he was a research associate in condensed matter theory at the University of California in Davis and at the University of Augsburg, Germany. He was a systems engineer with MTU Aero Engines, Munich, from 1998 to 2001. Since 2001 he is part of the scientific staff in the Department of Sensor Data and Information Fusion at the Fraunhofer FKIE where he is head of Research Group Wide Area Surveillance.



Wolfgang Koch studied physics and mathematics at Aachen Technical University (RWTH), Germany, where he earned a “Dr. rer. nat.” degree in theoretical physics. He is Head of the Department Sensor Data and Information Fusion at Fraunhofer FKIE, a German defence research lab. At Bonn University, he is teaching Applied Computer Science as a “Privatdozent.” Dr. Koch serves the IEEE AESS as an Associate Editor-in-Chief and Technical Editor of Target Tracking and Multisensor Systems for the IEEE Transactions on Aerospace and Electronic Systems. In 2013, he served as President of the International Society of Information Fusion (ISIF). In 2016, he was General Co-Chair of the IEEE ISF International Conference on Information Fusion, Heidelberg, Germany.

Applying Interacting Multiple Model to Financial Asset Allocation

SHOZO MORI
K. C. CHANG
HAJIME TAKAHASHI
CHEE-YEE CHONG

This paper describes a continuous-time state-process, discrete-time observation, Interacting Multiple Model (IMM) tracking algorithm, and its applications to financial market modeling and asset allocation. The system state is modeled as a continuous-time, affine-Gaussian stochastic dynamical process driven by a white process noise, as well as by structural changes modeled by a finite-state, continuous-time, Markov process. The system generally assumes multiple models with different state space dimensions, and an affine-Gaussian state jump whenever a model transition occurs. The underlying problem is a standard filtering problem for estimating the system state based on a sequence of discrete-time, linear-Gaussian observations of partial system states. To demonstrate the new method, we apply the IMM algorithm to financial market modeling for dynamic asset allocation. The resulting performance shows the potential applicability of the proposed method.

Manuscript received February 4, 2016; revised August 4, 2016; released for publication November 11, 2016.

Refereeing of this contribution was handled by Huimin Chen.

Authors' Addresses: S. Mori, Systems & Technology Research, Sunnyvale, CA, U.S.A. (E-mail: Shozo.Mori@STRResearch.com). K. C. Chang, Dept. of SEOR, George Mason University, Fairfax, VA, U.S.A. (E-mail: kchang@gmu.edu). H. Takahashi, Tottori University of Environmental Studies, Tottori, Japan (E-mail: hajime@kankyo-u.ac.jp). C.-Y. Chong, Independent Researcher, Los Altos, CA, U.S.A. (E-mail: cychong@ieee.org).

The research was partially supported by ARO under the grant #W911NF-15-1-0409.

1557-6418/17/\$17.00 © 2017 JAIF

1. INTRODUCTION

In this paper, we are generally concerned with financial market modeling and asset allocation problems, and specifically, with the possibility of applying Interacting Multiple Model (IMM) methods (which were developed as algorithms for tracking maneuvering targets [1] in 1980s, and since then, have been refined in many directions [3]) to financial market modeling. This paper expands the continuous-time IMM extrapolation algorithm introduced in [14] (which used a typical maneuvering target tracking example with stop-and-go target behavior as an illustration) to a full IMM tracking algorithm definition, and shows how the algorithm can be used to model financial market behaviors, as a continuous-time stochastic dynamical system with discrete-time observations, in which the system structure switches between multiple models.

Since the time when the IMM approach to tracking maneuvering targets was first published ([4, 5, 21]), the IMM methods have been widely used to make tracking algorithms adaptive to a wide range of target maneuvering and other abrupt structural changes in target motion dynamics. In fact, the IMM methods are one of the most studied subjects in target tracking, as documented in [3–7]. As a target tracking algorithm, each model used in an IMM algorithm typically represents a standard target behavior such as an almost-constant-velocity (called “nearly-constant-velocity” in [1]) model, and an almost-constant-rate turn model, or alternatively, multiple models may represent different levels of white process noises in the target dynamics so as to expand the range of tracking (filtering) bandwidth adaptively ([22]).

In a typical IMM implementation, both model switching and state transition are allowed to happen only on prescribed discrete time steps. Indeed, almost all the IMM literature starts with a discrete-time target dynamics formalism. As mentioned in [7], a few exceptions include [8, 24] in which the target dynamics are described by stochastic differential equations driven by Poisson processes (to model inter-model switching) as well as Wiener processes (to model intra-model diffusion). Those models are known as continuous-time Markov jump processes [2]. In contrast, the mathematical model used in this paper (first introduced in [14]) is expressed by a continuous-time Markov process on a hybrid state space explicitly through a semi-group of state transition operator and its infinitesimal generator. Like the model described in [9], our model allows switching across spaces with different dimensions, and as in [7, 10], our model allows the system state to jump whenever a model switching happens. These flexibilities have motivated us to explore the possibilities of applications to modeling of financial markets that exhibit similar behavior. As expressed in [17], our general motivation is to explore possibilities of applications of

engineering techniques to social and economic system analysis.

Recently, switching models have been proposed to analyze financial markets, as described in [18–20]. The application of IMM methods for such modeling is very natural and apparently straightforward. The use of the continuous time IMM may be appropriate because, for example, the stock prices change almost constantly during the day but many people only pay attention to the closing prices (when other detailed data also become unavailable). The multiple model approach, as shown in [18–20], typically uses two models, i.e., bull (up) and bear (down) models. The continuous-time IMM algorithm shown in this paper allows us to switch among the models with different dimensions. For that reason, we will use three models where the third model, “steady” model, has different (reduced) dimension, and test the applicability of our new IMM algorithm to a more flexible financial market model.

The purpose of this paper is to investigate the applicability of our continuous time IMM algorithm to financial market. We will demonstrate its performance with a popular benchmark equity market index (S&P 500 futures) on different time scales. While we believe this model applies to general market dynamics, its overall effectiveness is subject to additional future research and validation.

In the next section, Section 2, we will define a continuous-time jump Markov linear/affine system as a Markovian process on a hybrid state space, expressed as a formal direct sum of Euclidean spaces with generally different dimensions. We then define a filtering problem, a solution to which is given in Section 3, where an IMM algorithm, with continuous time extrapolation and discrete time updating, will be described. Section 4 shows a simple three-model financial market model with an IMM extrapolation algorithm. Numerical examples of financial market modeling and asset allocation analysis will be presented in Section 5, followed by the conclusions in Section 6.

The preliminary version of this paper was presented at the 18th International Conference on Information Fusion [23].¹ We have refined the conference paper and added derivations of the approximation-less calculation of the model probability and the first and second moments of the state probability distribution for each model for the extrapolation step, a major technical contribution of the paper outlined in [8, 14].

2. JUMP MARKOV MODEL

Consider M models, each of which is represented by a vector-matrix triple (A_m, b_m, B_m) that defines an Itô's linear or affine stochastic differential equation as $dx_t = (A_m x_t + b_m)dt + B_m dw_t$, $m \in \{1, \dots, M\}$, which defines a continuous-time stochastic process x_t on a Euclidean

space E_m , with a vector-valued, unit-intensity Wiener process w_t , on an appropriate time interval. Thus, within a model m , the state x_t is a Gaussian stochastic process such that each sample is continuous (no jump).

We assume that model transition is expressed by a continuous-time, $\{1, \dots, M\}$ -valued, time-homogeneous Markov process $(m_t)_{t \in [t_0, \infty)}$ with transition probability

$$P_h(m' | m) \stackrel{\text{def}}{=} \text{Prob}\{m_{t+h} = m' | m_t = m\} = \begin{cases} c_{mm'}h + o(h) & \text{if } m' \neq m \\ 1 - \sum_{\substack{m''=1 \\ m'' \neq m}}^M c_{mm''}h + o(h) & \text{otherwise} \end{cases} \quad (1)$$

for each $(m, m') \in \{1, \dots, M\}^2$, $h > 0$, and $t \in [t_0, \infty)$, with constants $c_{mm'} \geq 0$ for $m' \neq m$, $c_{mm} = -\sum_{m''=1, m'' \neq m}^M c_{mm''} < 0$, and a fixed initial time t_0 . We assume each model transition is accompanied by an affine-Gaussian jump. Namely, when a model transition from m to m' happens at time t , the target state jumps from $\lim_{h \downarrow 0} x_{t-h}$ in E_m to $x_t = \lim_{h \downarrow 0} x_{t+h}$ that is a generalized Gaussian random vector with mean vector $F_m^{m'} \lim_{h \downarrow 0} x_{t-h} + g_m^{m'}$ and a positive semi-definite covariance matrix $V_m^{m'}$, where $F_m^{m'}$, $g_m^{m'}$, and $V_m^{m'}$ are a vector and matrices with appropriate dimensions. We use the convention that $F_m^m = I$ (the identity matrix), $g_m^m = 0$ (the zero vector), and $V_m^m = 0$ (the zero matrix) for each m , thus preventing any jump within the same model.

A more precise mathematical model can be expressed as a continuous-time, time-homogeneous Markov process $(x_t, m_t)_{t \in [t_0, \infty)}$ on a hybrid state space³ $E \stackrel{\text{def}}{=} \bigcup_{m=1}^M E_m \times \{m\}$ that is a formal direct-sum of Euclidean spaces E_m with generally different dimensions, with a transition probability

$$\begin{aligned} & \text{Prob}\{x_{t+h_1+h_2} \in dx', m_{t+h_1+h_2} = m' | x_t = x, m_t = m\} \\ &= P_{h_1+h_2}(m' | m) \int_{E_m} \int_{E_{m'}} \mathcal{G}(dx'; \Delta F_m^{m'}(h_2)x'' \\ & \quad + \Delta g_m^{m'}(h_2), \Delta V_m^{m'}(h_2)) \mathcal{G}(dx''; F_m^{m'} x''' + g_m^{m'}, V_m^{m'}) \\ & \quad \mathcal{G}(dx'''; \Delta F_m^{m'}(h_1)x + \Delta g_m^{m'}(h_1), \Delta V_m^{m'}(h_1)) + o(h_1 + h_2) \end{aligned} \quad (2)$$

for each $(m, m') \in \{1, \dots, M\}^2$, each $x \in E_m$, each $t \in [t_0, \infty)$, and $h_1, h_2 > 0$, where,⁴ for each m and $h \geq 0$, $\Delta F_m^m(h) \stackrel{\text{def}}{=} e^{A_m h}$, $\Delta g_m^m(h) \stackrel{\text{def}}{=} \int_0^h e^{A_m \tau} b_m d\tau$, and $\Delta V_m^m(h) \stackrel{\text{def}}{=} \int_0^h e^{A_m \tau} Q_m e^{A_m^T \tau} d\tau$ with $Q_m = B_m B_m^T$. $\mathcal{G}(\cdot; \bar{\xi}, V)$ is the symbol for the generic generalized Gaussian distribution with mean vector $\bar{\xi}$ and a positive semi-definite covariance matrix V , of compatible dimensions, defined by its

²We assume the right-continuity to eliminate any ambiguity.

³Since $E = \mathbb{R}^n \times \{1, \dots, M\}$ if $E_m = \mathbb{R}^n$ for all $m \in \{1, \dots, M\}$, our choice of the state space provides a proper extension to the usual models used for multiple-model formulations, with $\mathbb{R} = (-\infty, \infty)$.

⁴By X^T we mean the transpose of a vector or a matrix X .

¹This conference paper received *Fusion 2015 Jean-Pierre Le Cadre Best Paper Award*.

characteristic function as

$$\int e^{\sqrt{-1}\zeta^T \xi} \mathcal{G}(d\xi; \bar{\xi}, V) = \exp\left(\sqrt{-1}\bar{\xi}^T \zeta - \frac{1}{2}\zeta^T V \zeta\right) \quad (3)$$

for each vector ζ with the dimension determined by the parameter pair (ξ, V) .

The discrete time observations, y_1, y_2, y_3, \dots , are modeled as

$$y_k = H_{m_{t_k} k} x_{t_k} + \eta_k \quad (4)$$

for each $k = 1, 2, 3, \dots$, with the time sequence, t_1, t_2, t_3, \dots , such that $t_0 \leq t_k < t_{k+1}$ for each k , with observation matrices, $(H_{mk})_{m=1}^M$, $k = 1, 2, 3, \dots$, of appropriate dimensions,⁵ and with zero-mean independent Gaussian vectors $\eta_1, \eta_2, \eta_3, \dots$, with covariance matrices⁶ $R_k = \mathbb{E}(\eta_k \eta_k^T)$. The independent initial condition at the initial time t_0 is given as,

$$\text{Prob}\{x_{t_0} \in dx, m_{t_0} = m\} = p_{m0} \mathcal{G}(dx; \bar{x}_{m0}, \bar{V}_{m0}) \quad (5)$$

with an initial model probability p_{m0} , mean vector \bar{x}_{m0} , and positive definite covariance matrix \bar{V}_{m0} , for each model $m \in 1, \dots, M$.

Then the filtering problem defined by eqns. (1) to (5) is the problem of characterizing the a posteriori probability distribution, expressed by $\hat{p}_{mk} = \text{Prob}\{m_{t_k} = m \mid y_1, \dots, y_k\}$ and $\text{Prob}\{x_{t_k} \in dx_{t_k} \mid m_{t_k} = m, y_1, \dots, y_k\}$ for each $m \in \{1, \dots, M\}$, and $k = 1, 2, 3, \dots$. It would be extremely difficult (if not impossible) to express $\text{Prob}\{x_{t_k} \in dx_{t_k} \mid m_{t_k} = m, y_1, \dots, y_k\}$ in any analytical (closed) form because of the infinitely many possibilities of how the system jumps occur, in any given interval $[t_{k-1}, t_k]$. However, as shown in the next section, the continuous-time evolution of the model probability \hat{p}_{mk} , and the first and the second moments of the posterior state probability distribution, $\text{Prob}\{x_{t_k} \in dx_{t_k} \mid m_{t_k} = m, y_1, \dots, y_k\}$, given model m , i.e., $\hat{x}_{mk} = \mathbb{E}(x_{t_k} \mid m_{t_k} = m, y_1, \dots, y_k)$ and $\hat{V}_{mk} = \mathbb{E}(x_{t_k} x_{t_k}^T \mid m_{t_k} = m, y_1, \dots, y_k) - \hat{x}_{mk} \hat{x}_{mk}^T$, can be analytically derived from eqns. (1) to (5), by a single vector homogeneous linear differential equation, as shown in the next section, Section 3.

Instead of modeling the continuous model switching by a stochastic differential equation driven by a Poisson process and a Wiener process, as formulated in [2, 8, 24], we have introduced a continuous-time Markov process on a hybrid space $\bigcup_{m=1}^M E_m \times \{m\}$, rather than $E = R^n \times \{1, \dots, M\}$, explicitly by a transition probability defined by (1) and (2), thereby extending the general continuous-time IMM models described in [7, 24, 25]. Moreover, we explicitly model any jump between the state spaces E_m and $E_{m'}$ with generally different dimensions, by a general affine jump, $x_{m'}' = F_m^{m'} x_m + g_m^{m'} + (V_m^{m'})^{1/2} \xi_m^{m'}$, from model m to m' , with zero-mean unit-variance Gaussian random vector

⁵Such that $H_{mk} \in R^{d_k \times \dim(E_m)}$ for every $m \in \{1, \dots, M\}$ where d_k is the dimension of y_k , for every k .

⁶ \mathbb{E} is the symbol for the conditional and unconditional mathematical expectation operators.

$\xi_m^{m'}$ to represent uncertainty in the jump. By doing so, we avoid the bias issues addressed in [26], which arise when state spaces with different dimensions are handled by adding artificial zero state components and applying the standard IMM mixing algorithm mechanically.

3. IMM ALGORITHM

First we consider the extrapolation step, generally following [14]. To do so, we define a semi-group of linear functionals \mathcal{T}_h on the space \mathcal{C} of all the real-valued bounded continuous functions ϕ on the hybrid space E by, $\mathcal{T}_h \phi(x, m) = \mathbb{E}(\phi(x_{t+h}, m_{t+h}) \mid x_t = x, m_t = m)$ for each $(x, m) \in E$, $t \in [t_0, \infty)$ and $h \geq 0$. Since (x_t, m_t) is a time-homogeneous Markov process, the definition does not depend on t . Then the *infinitesimal generator* \mathcal{A} of \mathcal{T}_h can be defined as

$$\begin{aligned} \mathcal{A}\phi(x, m) &= \lim_{h \downarrow 0} h^{-1} (\mathcal{T}_h \phi(x, m) - \phi(x, m)) \\ &= \frac{\partial}{\partial x} \phi(x, m) (A_m x + b_m) + \frac{1}{2} \text{trace} \left(\frac{\partial^2}{\partial x^2} \phi(x, m) Q_m \right) \\ &\quad + \sum_{m'=1}^M c_{mm'} \int_{E_{m'}} \phi(x', m') \mathcal{G}(dx'; F_m^{m'} x + g_m^{m'}, V_m^{m'}) \end{aligned} \quad (6)$$

More precisely, when the limit $\lim_{h \downarrow 0} h^{-1} (\mathcal{T}_h \phi - \phi)$ exists in the sup-norm of \mathcal{C} , we say the functional ϕ belongs to the domain of \mathcal{A} , i.e., $\phi \in \text{Dom}(\mathcal{A})$, and the last expression of eqn. (6) is uniquely implied⁷ by eqns. (1) and (2). Then, for any $\phi \in \text{Dom}(\mathcal{A})$, we have [11]

$$\begin{aligned} \mathbb{E}(\phi(x_{t+h}, m_{t+h}) \mid (x_t, m_t)) &= \phi(x_t, m_t) + \mathbb{E} \left(\int_t^{t+h} \mathcal{A}\phi(x_\tau, m_\tau) d\tau \mid (x_t, m_t) \right) \end{aligned} \quad (7)$$

With the (unconditional) expectation of both sides of (7), under a regularity condition that allows us to interchange the state-space expectation and the time-integral, we have

$$\mathbb{E}(\phi(x_{t+h}, m_{t+h})) = \mathbb{E}(\phi(x_t, m_t)) + \int_t^{t+h} E(\mathcal{A}\phi(x_\tau, m_\tau)) d\tau \quad (8)$$

or $(d/dt)\mathbb{E}(\phi(x_t, m_t)) = \mathbb{E}(\mathcal{A}\phi(x_t, m_t))$.

Let us define $\bar{p}_{mk}(t) = \text{Prob}\{m_t = m \mid y_1, \dots, y_k\}$, $\bar{x}_{mk}(t) = \mathbb{E}(x_t \mid m_t = m, y_1, \dots, y_k) \bar{p}_{mk}(t)$, and $\bar{S}_{mk}(t) = \mathbb{E}(x_t x_t^T \mid m_t = m, y_1, \dots, y_k) \bar{p}_{mk}(t)$, for each $m \in \{1, \dots, M\}$. Then it follows from (1), (2), and (8), that, for each $t \in [t_k, t_{k+1}]$, with C defined as the $M \times M$ matrix whose (i, j) element is c_{ij} defined in (1),

$$[\bar{p}_{1k}(t) \dots \bar{p}_{Mk}(t)] = [\hat{p}_{1k} \dots \hat{p}_{Mk}] \exp(C(t - t_k)) \quad (9)$$

⁷See Appendix A for the derivation of (6) from (1) and (2).

$$\begin{aligned} \frac{d}{dt}\bar{x}_{mk}(t) &= A_m\bar{x}_{mk}(t) + b_m\bar{p}_{mk}(t) \\ &+ \sum_{m'=1}^M c_{m'm}(F_{m'}^m\bar{x}_{m'k}(t) + g_{m'}^m\bar{p}_{m'k}(t)) \end{aligned} \quad (10)$$

and

$$\begin{aligned} \frac{d}{dt}\bar{S}_{mk}(t) &= A_m\bar{S}_{mk}(t) + \bar{S}_{mk}(t)A_m^T + b_m\bar{x}_{mk}(t)^T \\ &+ \bar{x}_{mk}(t)b_m^T + Q_m\bar{p}_{mk}(t) \\ &+ \sum_{m'=1}^M c_{m'm}(F_{m'}^m\bar{S}_{m'k}(t)(F_{m'}^m)^T \\ &+ F_{m'}^m\bar{x}_{m'k}(t)(g_{m'}^m)^T + g_{m'}^m\bar{x}_{m'k}(t)^T(F_{m'}^m)^T \\ &+ (g_{m'}^m(g_{m'}^m)^T + V_{m'}^m)\bar{p}_{m'k}(t)) \end{aligned} \quad (11)$$

The initial conditions for (10) and (11) are given as $\bar{x}_{mk}(t_k) = \hat{x}_{mk}\hat{P}_{mk}$ and $\bar{S}_{mk}(t_k) = (\hat{V}_{mk} + \hat{x}_{mk}\hat{x}_{mk}^T)\hat{P}_{mk}$. Eqn. (9) is a well-known formula, while the derivation of eqns. (10) and (11) are given in Appendix B.

For each $t \in [t_k, t_{k+1}]$, let $\Xi_t = (\bar{p}_{mk}(t), \bar{x}_{mk}(t), \bar{S}_{mk}(t))_{m=1}^M$ and let φ be the function that arranges all the elements in Ξ_t into a vector in the N -dimensional Euclidean space,⁸ with $N = \sum_{m=1}^M (1 + \dim(E_m) + \dim(E_m) \cdot (\dim(E_m) + 1)/2)$. Then, since all the equations (9) to (11) are linear ordinary differential equations, we have

$$\varphi(\Xi_t) = \exp(D(t-t'))\varphi(\Xi_{t'}) \quad (12)$$

for any (t, t') such that $t_k \leq t' \leq t \leq t_{k+1}$, where D is an $N \times N$ matrix uniquely defined by eqns. (9) to (11), and can be calculated by any one of the known effective numerical methods.

Furthermore, if we assume $\bar{p}_{mk}(t) > 0$ for any $m \in \{1, \dots, M\}$ and $t \in [t_k, t_{k+1}]$, it follows from (9) to (11) that

$$\begin{aligned} \frac{d}{dt}\tilde{V}_{mk}(t) &= A_m\tilde{V}_{mk}(t) + \tilde{V}_{mk}(t)A_m^T + Q_m\bar{p}_{mk}(t) \\ &+ \sum_{m'=1}^M c_{m'm}(F_{m'}^m\tilde{V}_{m'k}(t)(F_{m'}^m)^T \\ &+ \bar{p}_{m'k}(t)(V_{m'}^m + \Delta_{m'}^m(t)\Delta_{m'}^m(t)^T)) \end{aligned} \quad (13)$$

with

$$\begin{aligned} \tilde{V}_{mk}(t) \stackrel{\text{def}}{=} \mathbb{E} \left(\left(x_t - \frac{\bar{x}_{mk}(t)}{\bar{p}_{mk}(t)} \right) \left(x_t - \frac{\bar{x}_{mk}(t)}{\bar{p}_{mk}(t)} \right)^T \right. \\ \left. \middle| m_t = m, y_1, \dots, y_k \right) \bar{p}_{mk}(t) \end{aligned} \quad (14)$$

⁸We only need the values for the upper triangle elements for each symmetric matrix $\bar{S}_{km}(t)$.

and

$$\begin{aligned} \Delta_{m'}^m(t) \stackrel{\text{def}}{=} \bar{p}_{mk}(t)^{-1}\bar{x}_{mk}(t) - \bar{p}_{m'k}(t)^{-1}F_{m'}^m\bar{x}_{m'k}(t) \\ - g_{m'}^m = \bar{x}_{mk}(t) - (F_{m'}^m\bar{x}_{m'k}(t) + g_{m'}^m) \end{aligned} \quad (15)$$

We should note that, in (13) to (15), we have $V_m^m = 0$ and $\Delta_m^m = 0$, for each m .

The IMM update step, which precedes each extrapolation step described above, is performed by the standard IMM update formula. Namely, for each $m \in \{1, \dots, M\}$, assuming $\bar{p}_{m(k-1)}(t_k) > 0$, we have

$$\hat{x}_{mk} = \frac{\bar{x}_{m(k-1)}(t_k)}{\bar{p}_{m(k-1)}(t_k)} + K_{mk} \left(y_k - H_{mk} \frac{\bar{x}_{m(k-1)}(t_k)}{\bar{p}_{m(k-1)}(t_k)} \right) \quad (16)$$

$$\hat{V}_{mk} = (I - K_{mk}H_{mk})\bar{V}_{mk} \quad (17)$$

where

$$\bar{V}_{mk} = \frac{\bar{S}_{m(k-1)}(t_k)}{\bar{p}_{m(k-1)}(t_k)} - \left(\frac{\bar{x}_{m(k-1)}(t_k)}{\bar{p}_{m(k-1)}(t_k)} \right) \left(\frac{\bar{x}_{m(k-1)}(t_k)}{\bar{p}_{m(k-1)}(t_k)} \right)^T \quad (18)$$

and

$$K_{mk} = \bar{V}_{mk}H_{mk}^T S_{mk}^{-1} \quad (19)$$

with

$$S_{mk} = H_{mk}\bar{V}_{mk}H_{mk}^T + R_k \quad (20)$$

$$\hat{P}_{mk} = \left(\sum_{m'=1}^M L_{m'k} \right)^{-1} L_{mk} \quad (21)$$

and

$$L_{mk} = \frac{\bar{p}_{m(k-1)}(t_k)}{\sqrt{\det(2\pi S_{mk})}} \exp \left(-\frac{1}{2} \left\| y_k - \frac{\bar{x}_{m(k-1)}(t_k)}{\bar{p}_{m(k-1)}(t_k)} \right\|_{S_{mk}^{-1}}^2 \right) \quad (22)$$

The matrix H_{mk} in eqns. (16)–(20) is the observation matrix and R_k is the covariance matrix of the observation noise η_k , both used to define the measurement equation (4).

A critical step to develop a very simple solution in the form of the linear ordinary differential eqn. (12) is our use of the particular form of the first and the second moments, $\bar{x}_{mk}(t)$ and $\bar{S}_{mk}(t)$, rather than a usual choice of conditional mean and covariance, $\mathbb{E}(x_t | m_t, y_1, \dots, y_k)$ and $\mathbb{E}(x_t x_t^T | m_t, y_1, \dots, y_k) - \mathbb{E}(x_t | m_t, y_1, \dots, y_k)\mathbb{E}(x_t | m_t, y_1, \dots, y_k)^T$. To the best of our knowledge, this fact was shown in [8] for the first time, and expanded to a general multiple-model, affine-Gaussian dynamics and jumps in [14].

4. A SIMPLE FINANCIAL MARKET MODEL

As mentioned earlier, we will model the financial market dynamics with a simple multiple-model switching system, as in [18–20]. We use three models (i.e., $M = 3$), (i) “up” (“bull”), (ii) “steady,” and (iii) “down” (“bear”) models. Generally, we use “ u ” to rep-

represent the “price” in an appropriate sense, and “ v ” to represent its time derivative. The three models are defined as follows:

(i) Up (Bull) Model ($m = 1$) is based on a biased Ornstein-Uhlenbeck process, defined by the affine stochastic differential equation,

$$\begin{cases} du_t = v_t dt \\ dv_t = -\beta_1(v_t - \bar{v}_1)dt + \sqrt{q_1}dw_t \end{cases} \quad (23)$$

with unit-intensity Wiener process w_t , and three strictly positive parameters, $(\bar{v}_1, \beta_1, q_1)$.

(ii) Steady Model ($m = 2$) is a one-dimensional stationary stochastic process defined by

$$du_t = -\beta_0(u_t - \bar{u}_0)dt + \sqrt{q_0}dw'_t \quad (24)$$

with unit-intensity Wiener process w'_t , and three strictly positive parameters, $(\bar{u}_0, \beta_0, q_0)$.

(iii) Down (Bear) Model ($m = 3$) is another biased Ornstein-Uhlenbeck process defined by

$$\begin{cases} du_t = v_t dt \\ dv_t = -\beta_1(v_t + \bar{v}_1)dt + \sqrt{q_1}dw''_t \end{cases} \quad (25)$$

also with unit-intensity Wiener process w''_t . We can have a different set of parameters but will use the same set of parameters of Model 1 for simplicity.

Thus we have

$$A_1 = A_3 = \begin{bmatrix} 0 & 1 \\ 0 & -\beta_1 \end{bmatrix}, \quad b_1 = \begin{bmatrix} 0 \\ \beta_1 \bar{v}_1 \end{bmatrix} = -b_3,$$

$$B_1 = B_3 = \begin{bmatrix} 0 \\ \sqrt{q_1} \end{bmatrix}, \quad Q_1 = Q_3 = \begin{bmatrix} 0 & 0 \\ 0 & q_1 \end{bmatrix},$$

$A_2 = [-\beta_0]$, $b_2 = [\beta_0 \bar{u}_0]$, $B_2 = [\sqrt{q_0}]$, and $Q_2 = [q_2]$ with $E_1 = E_3 = (-\infty, \infty)^2$ and $E_2 = (-\infty, \infty)$. With symmetry assumption, the transition probabilities of eqn. (1) are defined by

$$C = \begin{bmatrix} -c_1 & c_1 & 0 \\ c_2/2 & -c_2 & c_2/2 \\ 0 & c_1 & -c_1 \end{bmatrix} \quad (26)$$

with two parameters, $c_1 > 0$ and $c_2 > 0$. $F_1^2 = F_3^2 = [1 \ 0]$, $g_1^2 = g_3^2 = V_1^2 = V_3^2 = [0]$,

$$F_2^1 = F_3^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad g_2^1 = \begin{bmatrix} 0 \\ \bar{v}_1 \end{bmatrix} = -g_3^1, \quad \text{and}$$

$$V_2^1 = V_3^1 = \begin{bmatrix} 0 & 0 \\ 0 & \bar{\sigma}_1^2 \end{bmatrix}, \quad \text{with } q_1 = 2\beta_1 \sigma_1^2.$$

Then we can write eqn. (12) explicitly as

$$\frac{d}{dt} \Xi(t) = \begin{bmatrix} D_{11} & 0 & 0 \\ D_{21} & D_{22} & 0 \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \Xi(t) \quad (27)$$

with $\Xi(t) = [\bar{p}_{1k}(t) \ \bar{p}_{2k}(t) \ \bar{p}_{3k}(t) \ \bar{x}_k(t)^T \ \tilde{S}_k(t)^T]^T$, where $\bar{x}_k(t) = [\bar{x}_{1k}(t)^T \ \bar{x}_{2k}(t)^T \ \bar{x}_{3k}(t)^T]^T$, $\tilde{S}_k(t) = [\tilde{S}_{1k}(t)^T \ \tilde{S}_{2k}(t)^T \ \tilde{S}_{3k}(t)^T]^T$ (with the vector representations \tilde{S}_k and \tilde{S}_{mk} for the matrices \tilde{S}_k and \tilde{S}_{mk}), $D_{11} = C^T$,

$$\begin{aligned} D_{21} &= \begin{bmatrix} 0 & 0 & 0 \\ \beta_1 \bar{v}_1 & c_2 \bar{v}_1/2 & 0 \\ 0 & \beta_0 \bar{u}_0 & 0 \\ 0 & 0 & 0 \\ 0 & -c_2 \bar{v}_1/2 & -\beta_1 \bar{v}_1 \end{bmatrix}, \quad D_{22} = \begin{bmatrix} -c_1 & 1 & c_2/2 & 0 & 0 \\ 0 & -\beta_1 - c_1 & 0 & 0 & 0 \\ c_1 & 0 & -\beta_0 - c_2 & c_1 & 0 \\ 0 & 0 & c_2/2 & -c_1 & 1 \\ 0 & 0 & 0 & 0 & -\beta_1 - c_1 \end{bmatrix}, \\ D_{31} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ q_1 & c_2(\bar{v}_1^2 + \bar{\sigma}_v^2)/2 & 0 \\ 0 & q_0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & c_2(\bar{v}_1^2 + \bar{\sigma}_v^2)/2 & q_1 \end{bmatrix}, \quad D_{32} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \beta_1 \bar{v}_1 & 0 & c_2 \bar{v}_1/2 & 0 & 0 \\ 0 & 2\beta_1 \bar{v}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -c_2 \bar{v}_1/2 & \beta_2 \bar{v}_2 & 0 \\ 0 & 0 & 0 & 0 & 2\beta_2 \bar{v}_2 \end{bmatrix}, \quad \text{and} \\ D_{33} &= \begin{bmatrix} -c_1 & 2 & 0 & c_2/2 & 0 & 0 & 0 \\ 0 & -\beta_1 - c_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2\beta_1 - c_1 & 0 & 0 & 0 & 0 \\ c_1 & 0 & 0 & -c_2/2 & c_1 & 0 & 0 \\ 0 & 0 & 0 & c_2/2 & -c_1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\beta_1 - c_1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2\beta_1 - c_1 \end{bmatrix} \end{aligned}$$

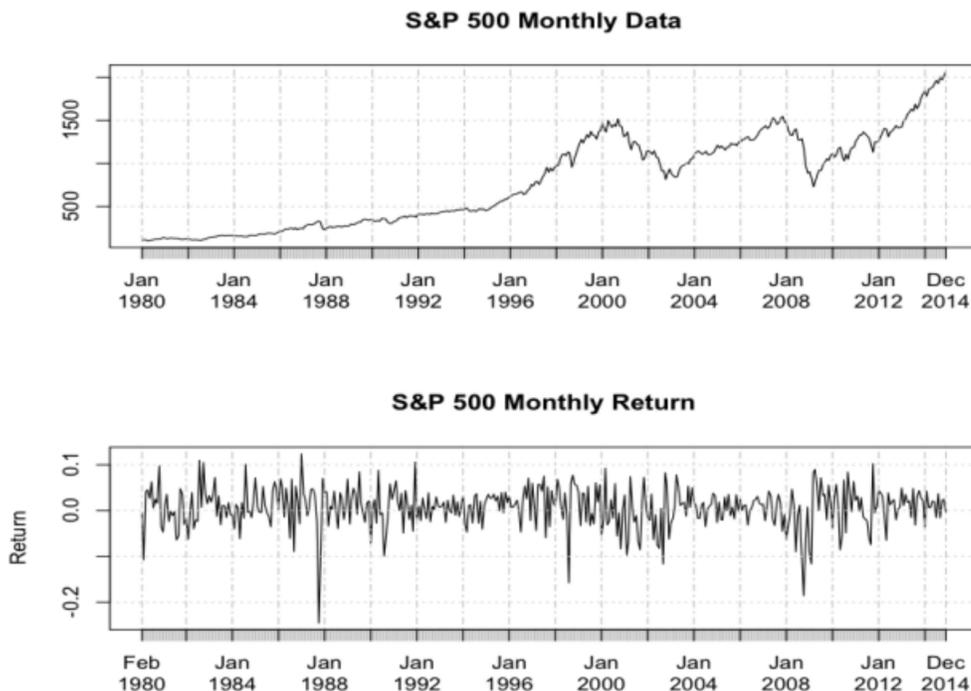


Fig. 1. Monthly S&P index form 1980 to 2014

Using the first measurement at time $t_1 = t_0$, the initial condition is given as $\bar{x}_{11}(t_1) = [y_1 \ \bar{v}_1]^T p_{10}$, $\bar{x}_{21}(t_1) = [y_1] p_{20}$, $\bar{x}_{31}(t_1) = [y_1 \ -\bar{v}_1]^T p_{30}$, $\bar{S}_{11}(t_1) = \text{diag}(R_1, \bar{\sigma}_{v_1}^2) \cdot p_{10} + \bar{x}_{11}(t_1)\bar{x}_{11}(t_1)^T/p_{10}$, $\bar{S}_{21}(t_1) = R_1 p_{20} + \bar{x}_{21}(t_1)^2/p_{20}$, and $\bar{S}_{31}(t_1) = \text{diag}(R_1, \bar{\sigma}_{v_1}^2) p_{30} + \bar{x}_{31}(t_1)\bar{x}_{31}(t_1)^T/p_{30}$, with the initial model probabilities $(p_{m0})_{m=1}^3$.

The measurement matrices are given by $H_{1k} = H_{3k} = [1 \ 0]$ and $H_{2k} = [1]$, for all $k = 1, 2, 3, \dots$

5. APPLYING IMM TO FINANCIAL MODELING FOR ASSET ALLOCATION

There are two main approaches to analyze financial markets for investment and portfolio management. Fundamental analysis considers economic factors to make subjective judgments on the qualitative relationship between portfolio and market returns, whereas technical analysis uses quantitative historical data to predict future price movement. In this paper, we use the technical analysis approach where we apply the IMM model described in the previous section to model the dynamics of the equity market based on historical data.

Specifically, we focused on modeling the Standard & Poor's 500 (S&P 500) index as well as how to dynamically allocate the asset to invest in the index futures according to the model prediction. S&P 500 index is an American stock index based on the combined capitalization of 500 large companies in the US. It is one of the most widely followed benchmarks for the US and the world economy. Figure 1 shows the S&P monthly historical data from 1980 to 2014.

To test the algorithm, we randomly selected one daily, one weekly, and one monthly data sets, each with

100 data points to evaluate the performance on different time scales accordingly. In order to assess the market condition, the closing prices were used as the measurements and the three dynamic models: "up (bull)," "steady," and "down (bear)" as described in the previous section were used to model the S&P dynamics. In each test, the resulting estimated probabilities of the three models from the IMM algorithm were used to make the asset allocation decisions. The parameters were set, without any significant adjustments, as⁹: $(p_{m0})_{m=1}^3 = (0.3, 0.5, 0.2)$, $c_1 = c_2 = 1/3 \text{ day}^{-1}$, $\beta_0 = \beta_1 = \beta_2 = 2 \text{ day}^{-1}$, $\sigma_0 = \$20$, $\sigma_1 = \sigma_2 = 2\$/\text{day}$, $\bar{u}_0 = \$1100$, and $\bar{v}_1 = -\bar{v}_2 = 4\$/\text{day}$.

Traditional investment strategies usually apply heuristic rules or numerical indicators obtained from the historical data to determine the market trends. For example, stochastic oscillator (SO) or relative strength index (RSI) are well-known financial momentum indicators for contrarian investing [16]. These indicators are designed to determine the market conditions such as a potential top (resistance) or a bottom (support). A contrarian investor buys and sells against the market sentiment during a specific time based on the indicators. In that sense, one could consider the IMM algorithm developed in this paper and the resulting model probabilities as another momentum indicator. This new indicator attempts to determine the potential market overbought or oversold conditions. For example, when the "up" model probability is the highest one among the three and is above a certain threshold, it may indicate an overbought condition, and when the "down" model probability is

⁹\$ represents S&P index (as a virtual unit).

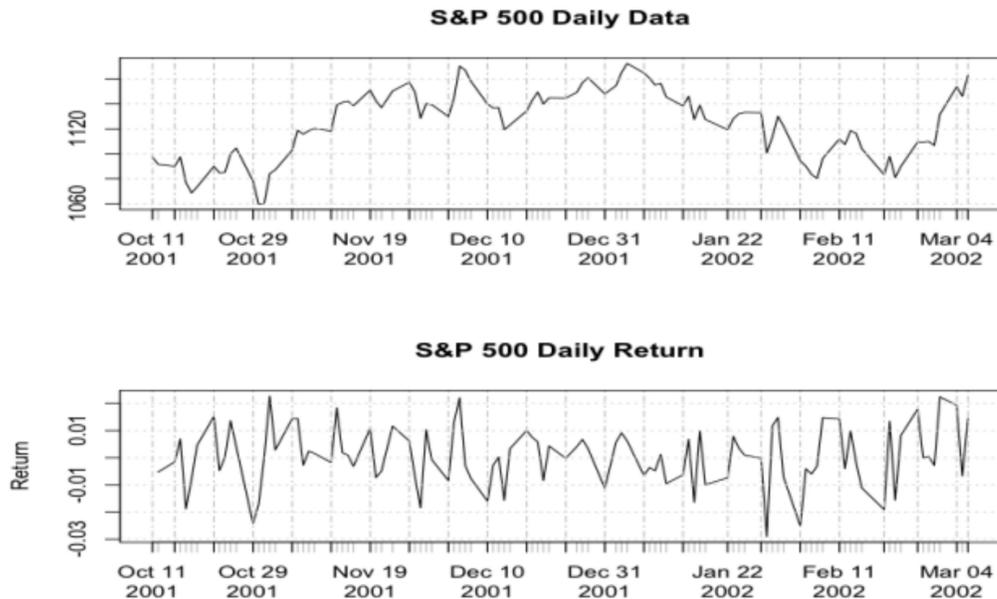


Fig. 2. S&P Daily Data—100 Days

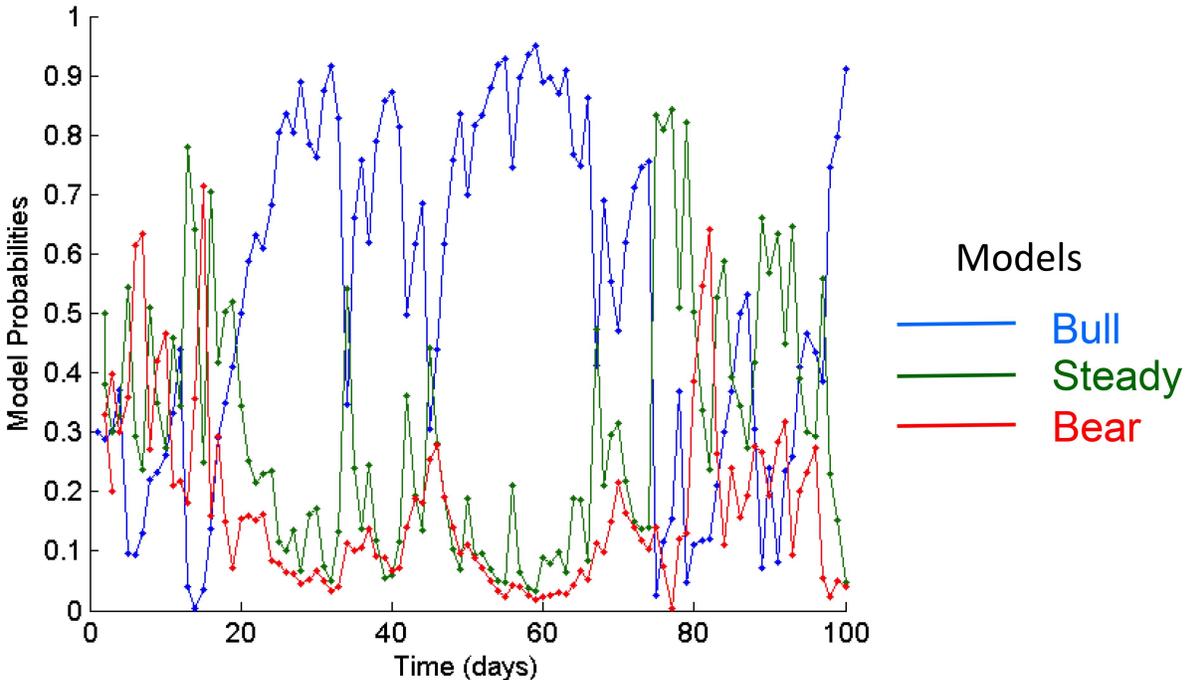


Fig. 3. IMM Model Probabilities—S&P Daily Data

the highest one and is above a certain threshold, it may indicate an oversold condition.

With the IMM indicator, we dynamically allocate the asset and make trading decisions accordingly. For example, a simple strategy is to short (sell) the S&P futures¹⁰ when the “up” probability is the highest one (overbought) and to long (buy) when the “down” probability is the highest one. We may also want to close

¹⁰S&P futures is one of the most liquid futures markets in the world. One can long or short the futures contracts as long as there is a counter party who is willing to take the opposite side.

our positions and sit on the sideline when the market is uncertain (“steady” mode probability is the highest). However, while this “contrarian” approach could lead to higher gain than usual, it may have the opposite effect when the market is in a strong trending mode. To mitigate this risk, when the IMM “up” or “down” probabilities are in extreme values (say, > 0.95) which indicates a potential strong trend, the decision rule mentioned above will be reversed to follow the market directions.

With the above simple asset allocation rules based on the IMM indicator, we conduct simulation and test their performances on the three randomly selected S&P data

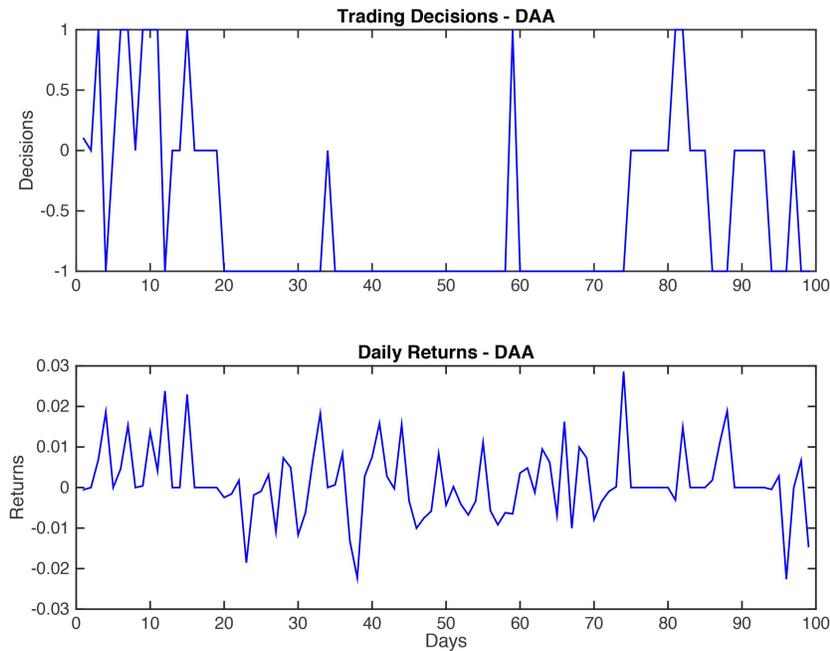


Fig. 4. IMM-DAA Trading Decisions and Daily Returns

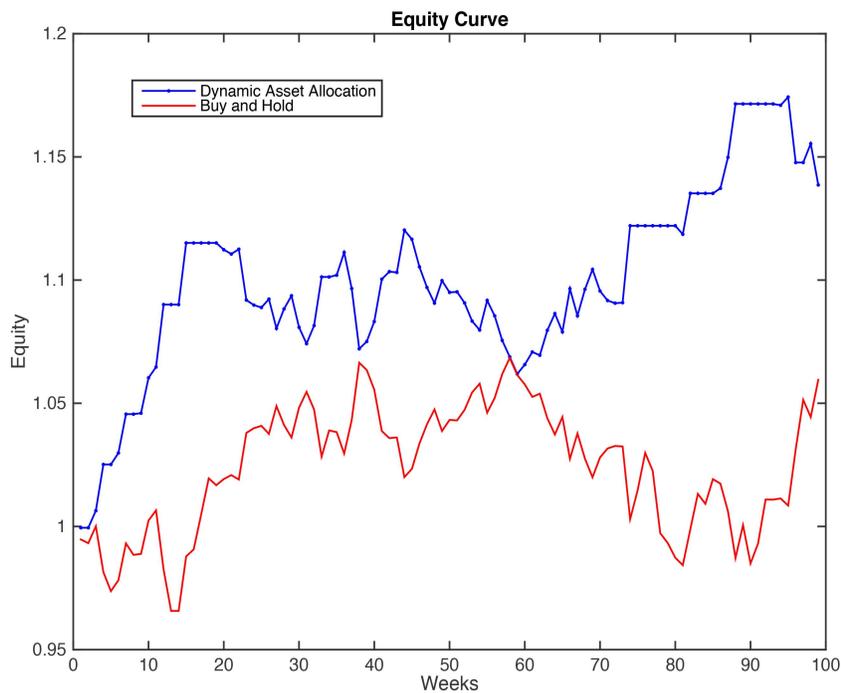


Fig. 5. Equity Curves—Buy-and-Hold vs. DAA

sets. We also compare its performance with the naïve buy-and-hold policy. Note that in the simulation, we use historical end-of-the-day S&P settlement prices to emulate the filled-prices of the transactions. We assume no transaction cost and no slippage.

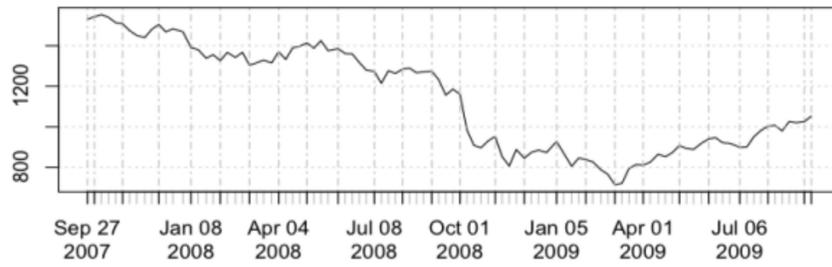
I. Daily Data

Figure 2 shows a randomly selected set of daily S&P closing prices and returns over a 100-day period. The daily returns represent the daily equity percentage

changes of the buy-and-hold strategy. Figure 3 shows the probability trajectories of the three models estimated by the IMM algorithm. In Figure 3, the model probabilities are shown by the blue line for the bull model ($m = 1$), the green line for the steady model ($m = 2$), and the red line for the bear model ($m = 3$).

The corresponding trading decisions of the IMM dynamic asset allocation (IMM-DAA) strategy and its daily returns are shown in Figure 4. In the figure, decision “1” represents a long position, “-1” represents

S&P 500 Weekly Data



S&P 500 Weekly Return

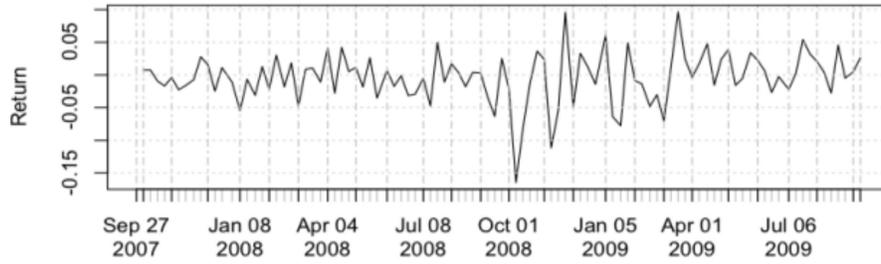


Fig. 6. S&P Weekly Data

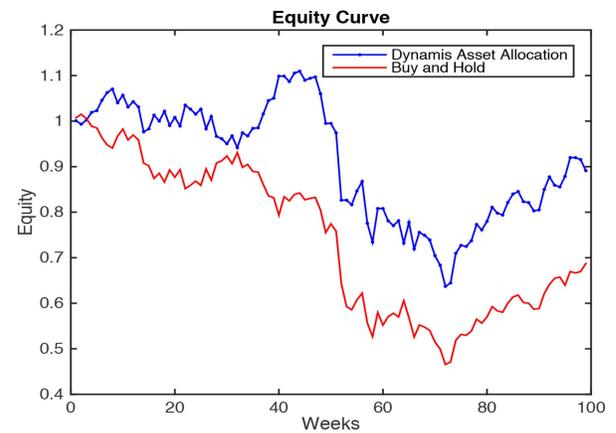
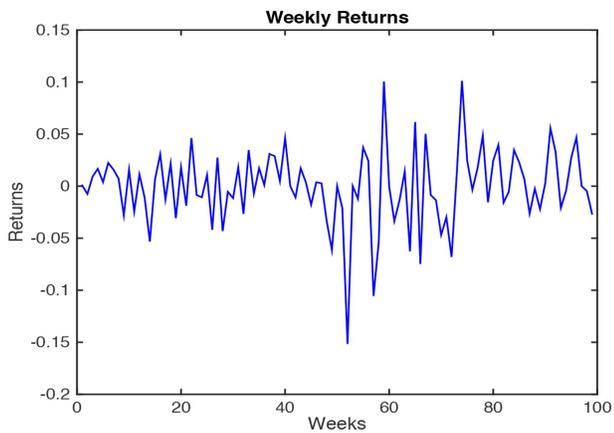
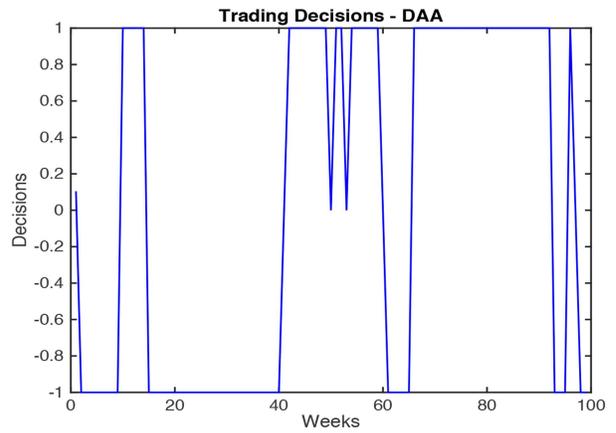


Fig. 7. Trading Performance—Weekly Data

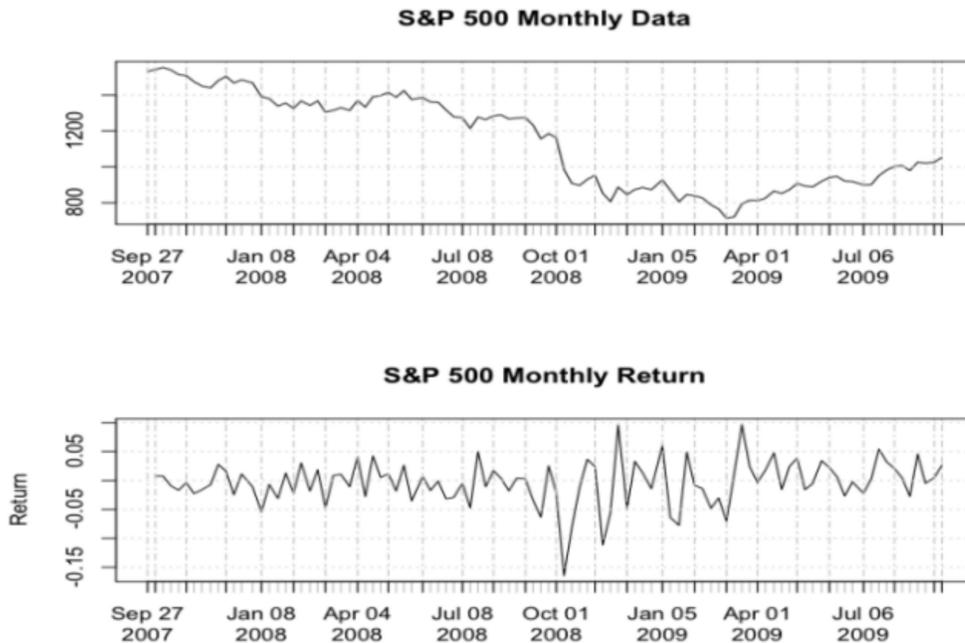


Fig. 8. S&P Monthly Data

a short position, and “0” represents no position. Figure 5 compares the equity curve over the 100-day period for the DAA strategy and the buy-and-hold (BH) policy. In the results, we assume no transaction costs or slippage.¹¹ As seen from the figure, DAA performs significantly better than the BH strategy with only a few trading actions—a total of around 20 over the 100-day period. At the end of the 100-day period, the cumulative return for BH is under 6% while DAA’s return is almost 14%. Note that the maximum drawdown¹² of the BH policy is approximately 7% while the maximum drawdown of the DAA is only about 5%.

In Figure 3, after the jump from the lower dimensional model to a higher dimensional model, the model probability history may have some appearance of the “bias,” which may be a result of a slight mismatch of the mean g_2^1 or g_2^3 with the real data.

II. Weekly and Monthly Data

Figures 6–9 show the results corresponding to the weekly and monthly data. Note that the decision rules based on the IMM indicators are exactly the same for the three data sets. Since we use the continuous-time system model, we do not have to adjust the system dynamics parameters in response to the sampling intervals, $t_{k+1} - t_k$. Based on the Markov property, the standard de-

viation of the process noise (volatility) is proportional to the square root of the time difference between two subsequent observations.

As shown in the figures, DAA either performs better than or close to BH with significantly lower drawdown. For example, Figure 7 shows that while BH loses about 31% of the equity over the 100 weeks period with a maximum drawdown of about 54%, DAA only loses 11% with a maximum drawdown of 43% over the same time period. Similarly, over a 100-month period, Figure 9 shows that while BH earns about 26% of the equity with a maximum drawdown of about 54%, DAA earns a slightly less return of 21% over the same period but with a significantly smaller drawdown of only 24%. Note that the randomly selected 100-month period includes the 2007–2008 credit crisis where prolonged market ups and downs exist for many months. While it is true that the rate of return and Sharpe ratio for BH are slightly better than that of DAA for this monthly time period, the maximum drawdown for BH is almost 230% higher than DAA which itself could be catastrophic. This demonstrates another potential benefit of applying the proposed DAA approach.

Table 1 summarizes the performance results for the three randomly selected data sets. In the table, an industry-standard performance indicator called the “Sharpe ratio”¹³ is also presented for performance comparison. Higher Sharpe ratio indicates a better risk-adjusted return. It is clear from the table that the IMM based DAA (IMM-DAA) is an effective and promising asset allocation method.

¹¹For S&P futures trading, given the liquidity and market size, the transaction cost is minimum. For example, with a standard e-mini S&P futures contract (~\$100k), the average transaction cost is less than 0.005% (< \$5) of the contract size. Given that in the 100 trading periods, there were about 20 transactions, the difference is negligible (~0.1%).

¹²Drawdown is defined as the peak-to-trough decline during a specific period of an investment. A drawdown is usually quoted as the percentage between the peak and the trough [16].

¹³The Sharpe ratio is a measure for calculating risk-adjusted return. It is the average return earned in excess of the risk-free rate over the return volatility (standard deviation).

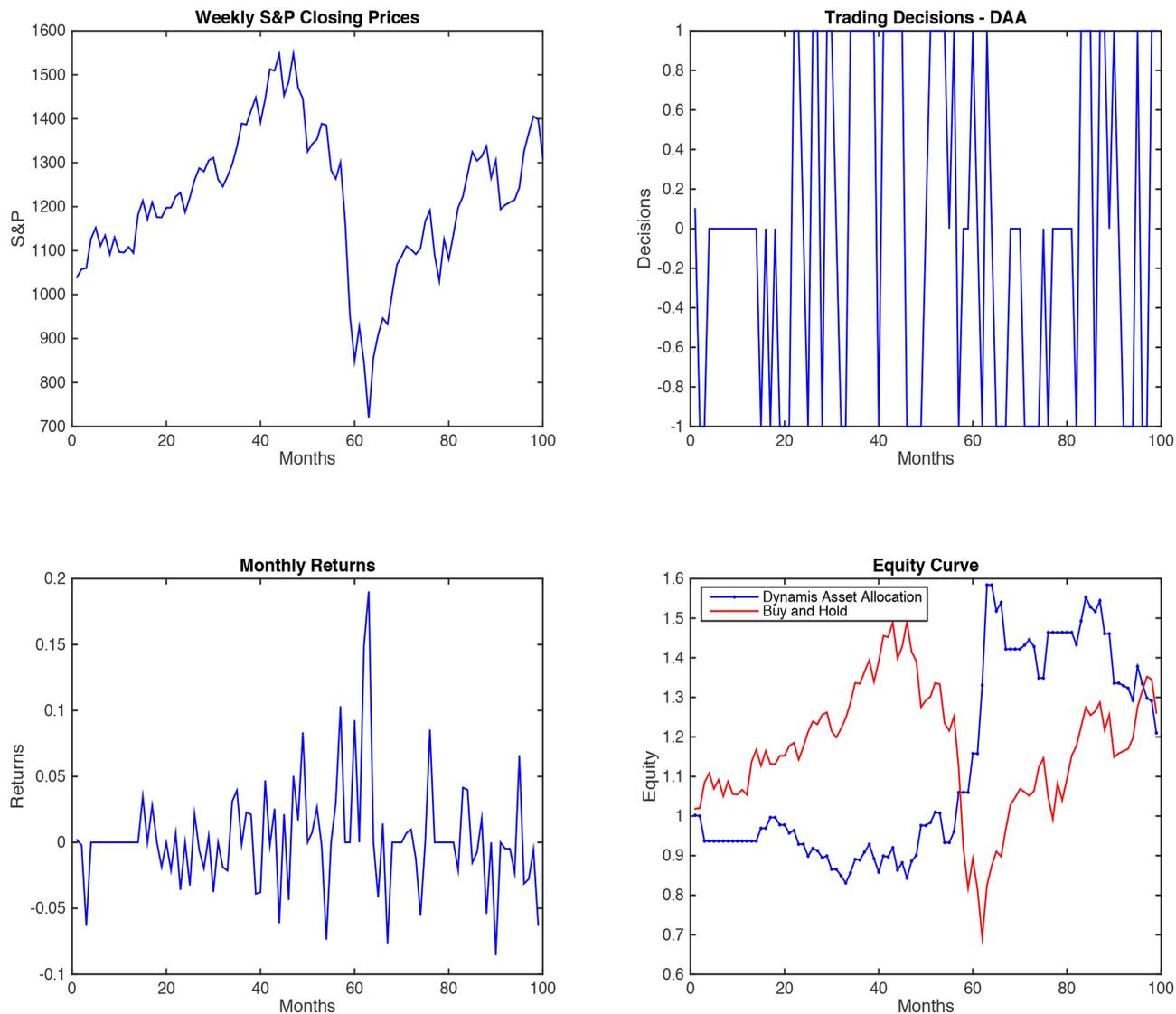


Fig. 9. Trading Performance—Monthly Data

6. CONCLUSIONS

In this paper, we presented a continuous-time, discrete-observation, Interacting Multiple Model (IMM) algorithm, based on the continuous-time IMM extrapolation developed in [14], and applied it to financial market dynamic modeling. We modeled the system by a continuous-time, jump Markov process and estimated the system state based on a sequence of discrete-time, linear-Gaussian observations. We utilized a rather naïve switching process with multiple linear stochastic system models to represent the S&P market dynamics model. The resulting IMM model probabilities serve as momentum indicators to make the dynamic asset allocation decisions (DAA). We tested the resulting IMM-DAA strategy on several randomly selected S&P data sets of various time scales. The results showed that the newly developed IMM indicator and the corresponding asset allocation strategy may have a potential to significantly

TABLE 1
Performance Comparison

	Rate of Return	Maximum Drawdown	Sharpe Ratio
Daily—BH	5.95%	7.88%	0.878
Daily—DAA	-13.86%	5.40%	2.307
Weekly—BH	-31.26%	54.14%	-0.615
Weekly—DAA	-10.88%	42.60%	-0.134
Monthly—BH	-26.08%	53.54%	0.192
Monthly—DAA	-21.00%	23.62%	0.160

outperform the baseline naïve buy-and-hold policy with lower risk.

The goal of this paper is to demonstrate the potential of the new continuous-time IMM algorithm when applied to financial market analysis and asset allocation problems. To demonstrate the effectiveness of the application of the new IMM algorithm to general financial market problems requires additional testing of large

amounts of real data and comparison with other methods proposed in the past, which is beyond the scope of this paper. Furthermore, in financial market modeling, consideration of non-Gaussian disturbance may be of significant interest. As seen in Sections 2 and 3, the development of our continuous-time IMM algorithm depends crucially on the Gaussian assumptions. A non-Gaussian extension of the proposed approach is therefore an interesting immediate sequel of the research of this paper.

Beyond several obvious refinements to the presented modeling approach, e.g., choosing the number of models, adjusting the intra and inter multiple model system parameters, adding ‘‘rate of model change’’ to the model probability itself, a potential future research direction to extend the promising preliminary work is to combine the technical approach described in this paper with a fundamental approach where both qualitative and quantitative information is utilized. Particularly, we should explore the highly relevant and emerging data fusion paradigm such as Bayesian networks and social networks for financial modeling and portfolio risk management.

APPENDIX A: CALCULATION OF INFINITESIMAL GENERATOR

For given $\phi \in \mathcal{C}$, it follows from (2) and the definition of the operator \mathcal{T}_h that

$$\begin{aligned} \mathcal{T}_{h_1+h_2}\phi(x, m) &= \sum_{m'=1}^M P_{h_1+h_2}(m' | m) \int_{E_{m'}} \phi(x', m') \\ &\quad \cdot \Phi_m^{m'}(dx'; x, h_1, h_2) + o(h_1 + h_2) \end{aligned} \quad (28)$$

for any $h_1, h_2 > 0$, each $m \in \{1, \dots, M\}$ and $x \in E_m$, where $\Phi_m^{m'}(\cdot; x, h_1, h_2)$ is the convolution of the three generalized Gaussian distributions in (2). Substituting (1) into (28), we have

$$\begin{aligned} \mathcal{T}_{h_1+h_2}\phi(x, m) &= \sum_{\substack{m'=1 \\ m' \neq m}}^M c_{mm'}(h_1 + h_2) \\ &\quad \times \int_{E_{m'}} \phi(x', m') \Phi_m^{m'}(dx'; x, h_1, h_2) \\ &\quad + \left(1 - \sum_{\substack{m'=1 \\ m' \neq m}}^M c_{mm'}(h_1 + h_2) \right) \\ &\quad \times \int_{E_m} \phi(x', m) \Phi_m^m(dx'; x, h_1, h_2) + o(h_1 + h_2) \end{aligned} \quad (29)$$

Hence, we have, for $h > 0$,

$$\begin{aligned} h^{-1}(\mathcal{T}_h\phi(x, m) - \phi(x, m)) &= \sum_{m'=1}^M c_{mm'} \int_{E_{m'}} \phi(x', m') \mathcal{G}_{m'}(dx'; F_m^{m'}x + g_m^{m'}, V_m^{m'}) \\ &\quad + h^{-1} \left(\int_{E_m} \phi(x', m) \mathcal{G}_m(dx'; e^{A_m h}x + b_m, \Delta V_m(h)) - \phi(x, m) \right) \\ &\quad + o(h) \end{aligned} \quad (30)$$

It is well known (e.g., cf. [11]) that the second term of the right hand side converges to $(\partial/\partial x)\phi(x, m)(A_m x + b_m) + \frac{1}{2}\text{trace}((\partial^2/\partial x^2)\phi(x, m)Q_m)$, and (5) follows.

For a fixed pair (h_1, h_2) , eqn. (2) implies the model transition from m to m' happens at most one time in the time interval $[t, t + h_1 + h_2]$ at time $t + h_1$. Usual IMM practice (e.g., cf. [4] or [6]) is to let $h_1 = 0$ and use a time interval $h = h_2$ that is equal to the sensor revisit time, and to use a Gaussian approximation.¹⁴ In [14], a multiple-model extrapolation algorithm where two or more model transitions are possible within a given extrapolation time interval was developed analytically without sub-dividing the extrapolation interval, which inevitably involves Gaussian approximation for each subinterval. Instead, the extrapolation algorithm developed in [14] and described in Appendix B preserves exact moment calculations by (8) and (10). At the end of the extrapolation interval, however, we need a Gaussian approximation to apply the IMM updating step, as seen in Section 3.

APPENDIX B: MOMENT CALCULATIONS

For a fixed $m \in \{1, \dots, M\}$ and a fixed $i \in \{1, \dots, \dim(E_m)\}$, define ϕ by $\phi(x, m') = x_i$ if $m' = m$, 0 otherwise. Then substituting this ϕ into eqn. (5), we have

$$\mathcal{A}\phi(x_t, m_t) = \delta_{m,m}(A_m x_t + b_m)_i + c_{m,m}(F_m^m x_t + g_m^m)_i \quad (31)$$

for each t . Taking expectation of (31) leads to

$$\begin{aligned} \mathbb{E}(\mathcal{A}\phi(x_t, m_t)) &= \sum_{m'=1}^M \mathbb{E}(\mathcal{A}\phi(x_t, m_t) | m_t = m') \text{Prob}\{m_t = m'\} \\ &= \sum_{m'=1}^M p_t(m') \mathbb{E}(\delta_{m',m}(A_m x_t + b_m)_i \\ &\quad + c_{m',m}(F_{m'}^m x_t + g_{m'}^m)_i | m_t = m') \\ &= p_t(m)(A_m \bar{x}(t | m) + b_m)_i + \sum_{m'=1}^M p_t(m') c_{m',m} (F_{m'}^m \bar{x}(t | m') + g_{m'}^m)_i \\ &= (A_m \bar{x}(t; m) + b_m p_t(m))_i + \sum_{m'=1}^M c_{m',m} (F_{m'}^m \bar{x}(t; m') + g_{m'}^m p_t(m'))_i \end{aligned} \quad (32)$$

¹⁴In the IMM literature, this Gaussian approximation is often referred to as *mixing*, which is also characterized as *interacting* among multiple models.

from which eqn. (10) follows, with $p_t(m') = \text{Prob}\{m_t = m'\}$, $x(t | m') = \mathbb{E}(x_t | m_t = m')$ and $\dot{x}(t; m') = \dot{x}(t | m') \times p_t(m')$, for every $m' \in \{1, \dots, M\}$.

The ordinary differential equation (11) for the non-centric second moments, $S(t; m) = S(t | m)p_t(m)$ with $S(t | m) = \mathbb{E}(x_t x_t^T | m_t = m)$, can be obtained in a similar way, using ϕ defined as, for a fixed $m \in \{1, \dots, M\}$ and a pair (i, j) such that $(i, j) \in \{1, \dots, \dim(E_m)\}^2$, $\phi(x, m') = x_i x_j$ if $m' = m$, zero otherwise.

In order to obtain eqn. (13), we should first note $S(t; m) = V(t; m) + \bar{x}(t; m)\bar{x}(t; m)^T p_t(m)^{-1}$, which implies

$$\begin{aligned} \dot{S}(t; m) &= \dot{V}(t; m) - \bar{x}(t; m)\bar{x}(t; m)^T p_t(m)^{-2} \dot{p}_t(m) \\ &\quad + (\dot{\bar{x}}(t; m)\bar{x}(t; m)^T + \bar{x}(t; m)\dot{\bar{x}}(t; m)^T) p_t(m)^{-1} \end{aligned} \quad (33)$$

with $\dot{S}(t; m) = (d/dt)S(t; m)$, $\dot{V}(t; m) = (d/dt)V(t; m)$, $\dot{\bar{x}}(t; m) = (d/dt)\bar{x}(t; m)$, and $\dot{p}_t(m) = (d/dt)p_t(m) = \sum_{m'=1}^M p_t(m')c_{m'm}$. Then, eqn. (13) is obtained by substituting eqns. (10), (11) and (15) into (33).

We should note that in order to derive the first and the second moments through eqns. (10) and (11), to be precise, we need one highly technical step, because, for example, $\phi(x, m) = x_i$ if $m = m'$, 0 otherwise, does not define a bounded functional ϕ on $\bigcup_{m=1}^M E_m \times \{m\}$. To justify the use of eqns. (6) to (8), we may need to consider a series of stopped processes, each bounded by a compact set $\{(x, m) \mid \|x\| \leq k\}$ for each integer k , and to apply Dynkin's lemma to obtain the desired result as a limit, as is done in [12] and [13].

REFERENCES

- [1] Y. Bar-Shalom, P. K. Willet, and X. Tian
Tracking and Data Fusion: A Handbook of Algorithms
YBS Publishing, 2011.
- [2] O. L. V. Costa, M. D. Fragoso, and M. G. Todoró
Continuous-Time Markov Jump Linear Systems,
Springer, 2012.
- [3] X.-R. Li, and V. P. Jilkov
Survey of Maneuvering Target Tracking, Part V: Multiple-Model Methods
IEEE Transactions on Aerospace and Electronic Systems, 41, 4, (October 2005), 1255–1321.
- [4] H. A. P. Blom
A Sophisticated Tracking Algorithm for ATC Surveillance Data
In *Proceeding of International Radar Conference*, Paris, France, May 1984.
- [5] H. A. P. Blom
An Efficient Filter for Abruptly Changing Systems
In *Proceedings of 23rd Conference on Decision and Control*, Las Vegas, NV, December 1984, 656–658.
- [6] X.-R. Li
Engineer's Guide to Variable-Structure Multiple-Model Estimation for Tracking
In *Multitarget-Multisensor Tracking: Applications and Advances*, Vol. III, ed. by Yaakov Bar-Shalom and William Dale Blair, Chap. 10, Artech House, 2000, 499–567.
- [7] H. A. P. Blom
The Continuous Time Roots of the Interacting Multiple Model Filter
In *Integrated Tracking, Classification, and Sensor Management, Theory and Applications*, ed. by Mahendra Mallick, et al., IEEE Press, 2013, 127–161.
- [8] J. W. Adaska
Mover-Sitter Analysis Using Stochastic Differential Equation Driven by Poisson Counters
Technical Report, BAE Systems, Advanced Information Technologies, TR 2193, Burlington, MA, July 2007.
- [9] Y. Bar-Shalom, and K. Birmiwal
Variable Dimension Filter for Maneuvering Target Tracking
IEEE Transactions on Aerospace and Electronic Systems, 18, 5, September 1982, 621–628.
- [10] H. A. P. Blom
Overlooked Potential of System with Markovian Coefficients
In *Proceedings of 25th Conference on Decision and Control*, Athens, Greece, December 1986, 1758–1764.
- [11] K. Itô
Essentials of Stochastic Processes
(translation of the original book, Kakuritsu Katei Ron, in Japanese, Iwanami Shoten, 1957) Translation of Mathematical Monograph Series, Vol. 231, American Mathematical Society, 2006.
- [12] H. J. Kushner
Stochastic Stability and Control
Academic Press, 1967.
- [13] E. B. Dynkin
Markov processes
Springer, 1965.
- [14] S. Mori, J. W. Adaska, M. A. Pravia, and C.-Y. Chong
Continuous-Time Interacting Multiple Model Extrapolation
In *Proceedings of 11th International Conference on Information Fusion*, Cologne, Germany, July 2008.
- [15] Yahoo Finance
<http://finance.yahoo.com>.
- [16] Technical Indicators
<http://www.investopedia.com/terms/t/technicalindicator.asp>.
- [17] D. G. Luenberger
Investment Science
Oxford University Press, 1998.
- [18] M. Guidolin
Markov Switching Models in Empirical Finance
Working Paper 415, IGIER, Università Bocconi, Milano, Italy, 2012.
- [19] C.-J. Kim, and C. R. Nelson
State-Space Models with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications
MIT Press, 1999.
- [20] J. M. Maheu, T. H. MaCurdy, and Y. Song
Components of Bull and Bear Markets: Bull Corrections and Bear Rallies
Journal of Business & Economic Statistics, Taylor & Francis Journals, 30, 3 February 2012, 391–403.
- [21] H. A. P. Blom, and Y. Bar-Shalom
The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients
IEEE Transactions on Automatic Control, 33, 8, August 1988, 780–783.
- [22] G. A. Watson, and W. D. Blair
IMM Algorithm for Tracking Targets That Maneuver through Coordinated Turns
In *Proceedings of SPIE Symposium on Signal and Data Processing of Small Targets Date Published*, Orlando, FL, August 1992.

- [23] S. Mori, K. C. Chang, H. Takahashi, and C.-Y. Chong
An Application of Interacting Multiple Model Tracking Method to Financial Modeling and Asset Allocation
In *Proceedings of 18th International Conference on Information Fusion*, Washington DC, U.S.A., July 2015.
- [24] H. A. P. Blom
Markov jump-diffusion and decision-making-free filtering
In *Proceedings of 6th International Conference Analysis and Optimization of Systems, Part 1*, ed. by Alain Bensoussan, and Jacques-Louis Lions, pp. 568–580, Springer, June 1984.
- [25] H. A. P. Blom
Continuous-Discrete Filtering for Systems with Markovian Switching Coefficients and Simultaneous Jumps
In *Proceedings of 21st Asilomar Conference on Signals, Systems, and Computation*, Pacific Grove, 1987.
- [26] T. Yuan, Y. Bar-Shalom, P. Willett, E. Mozeson, S. Pollak, and D. F. Hardiman
A Multiple IMM Estimation Approach with Unbiased Mixing for Thrusting Projectiles
IEEE Transactions on Aerospace and Electronic Systems, 48, 4, October 2012, 3250–3267.



Dr. Shozo Mori is a Senior Member of Technical Staff, at Systems & Technology Research (STR), at Sunnyvale, CA, office. In 2003 to 2012, he was with BAE Systems, Advanced Information Technologies Division, formerly ALPHATECH. Before joining ALPHATECH in 2003, he was with Information Extraction & Transport, Arlington, VA, Raytheon Company (Tiburon Systems), San Jose, CA, and Advanced Decision Systems (Advanced Information & Decision Systems), Mountain View, CA.

He obtained the B.S. in Electric Engineering from Waseda University, Tokyo, Japan, the M.S. in Control Engineering from Tokyo Institute of Technology, Tokyo, Japan, and the Ph.D. in Engineering–Economic Systems from Stanford University, Stanford, CA.

His current main research interest is in multiple-object tracking, particularly, as applications of theories of random finite sets, finite point processes, and distributed state estimation, as well as applications of theory of Markov jump processes to economic and social system analysis.



Dr. Kuo-Chu Chang received the M.S. and Ph.D. degrees in Electrical Engineering from the University of Connecticut in 1983 and 1986 respectively. From 1983 to 1992, he was a senior research scientist in Advanced Decision Systems (ADS) division, Booz-Allen & Hamilton, Mountain View, California. In 1992, he joined the Systems Engineering and Operations Research department, George Mason University where he is currently a professor. His research interests include estimation theory, optimization, signal processing, and multisensor data fusion. He is particularly interested in applying unconventional techniques in the conventional decision and control systems. He has more than 30 years of industrial and academic experience and published more than two hundred papers in the areas of multitarget tracking, distributed sensor fusion, Bayesian Networks technologies, and financial engineering. He was an associate editor on Tracking/Navigation Systems from 1993 to 1996 and on Large Scale Systems from 1996 to 2006 for IEEE Transactions on Aerospace and Electronic Systems. He was also an associate editor of IEEE Transactions on Systems, Man, and Cybernetics, Part A, from 2002 to 2007. He was the Technical Program Co-chair of the 2009 International Conference on Information Fusion, Seattle, USA.



Dr. Hajime Takahashi is President of Tottori University of Environmental Studies, Tottori, Japan. His teaching career includes: the University of Michigan, Ann Arbor, MI, from 1977 to 1979; Boston University, Boston, MA, from 1979 to 1982; Toyama University, Toyama, Japan, from 1982 to 1985; and Hitotsubashi University, Tokyo, Japan, from 1985 to 2012, where he was Dean of School of Economics, from 2001 to 2003, and Director of Computer Center, from 2003 to 2004. He also served as President of Japanese Association Finance Economic and Engineering, from 2005 to 2007.

Dr. Takahashi received his B.A. and M.A. degrees in Economics from Hitotsubashi University, Tokyo, Japan, and M.A. and Ph.D. degrees in Mathematical Statistics from Columbia University, New York, NY.

His current research interest includes statistical analysis of implied data in financial economics, and the theoretical aspects of the bootstrap method.



Dr. Chee-Yee Chong is an independent researcher in Los Altos, California, USA. He was on the electrical engineering faculty of Georgia Institute of Technology, Atlanta, Georgia, until 1980, when he moved to California. From 1980 to 2013, he was with Advanced Decision Systems (Mountain View, CA), which was acquired by Booz Allen Hamilton in 1991, and ALPHATECH (with headquarters in Boston area), which was acquired by BAE Systems in 2004. He obtained his bachelor, master, and doctoral degrees, all in electrical engineering, from Massachusetts Institute of Technology.

His research interests include signal processing, estimation, optimization, probabilistic reasoning, graphs, and machine learning. Current applications include multiple object tracking and data fusion, with emphasis on distributed implementations for large networks.

Tracking a Maneuvering Target Using Two Heterogeneous Passive Sensors on a Single Stationary Platform with IMM Estimation

HONG AN JACK HUANG
YAAKOV BAR-SHALOM
RONG YANG
GEE WAH NG

Bearing-only passive sensors have the advantage of being non-detectable, but they come with target state observability limitations. A new approach, the unscented Gauss-Helmert filter that fuses out-of-sequence acoustic measurements (OOSM-A) and electro-optic (EO) measurements (OOSM-AE), has been developed recently to fuse non-delayed and delayed measurements from two heterogeneous passive sensors on a single platform to overcome these observability issues. In this paper, we extend the OOSM-AE approach to use interacting multiple models (IMM) to improve target tracking accuracy when tracking maneuvering targets. The maneuvers considered are circular motion and S-turns. The resulting IMMOOSM-AE handles the delayed acoustic measurements as out-of-sequence measurements. Scenarios are simulated and tested with both IMMOOSM-AE and OOSM-AE and results are presented.

Manuscript received July 25, 2016; revised November 4, 2016; released for publication December 15, 2016.

Refereeing of this contribution was handled by Paolo Braca.

Authors' addresses: H. A. J. Huang, R. Yang and G. W. Ng, DSO National Laboratories, 12 Science Park Drive, Singapore 118225 (E-mail: hhongan@dso.org.sg; yrong@dso.org.sg; ngeewah@dso.org.sg). Y. Bar-Shalom, Department of ECE, University of Connecticut, Storrs, CT 06269, USA (E-mail: ybs@enr.uconn.edu).

Supported by ARO Grant W911NF-10-1-0369.

1557-6418/17/\$17.00 © 2017 JAIF

I. INTRODUCTION

There are operational merits to using passive sensors. Passive sensors are usually covert and non-detectable. However, range information is usually not available from these sensors. It then becomes challenging to initiate and track a target from a single passive sensor. This is known as the bearing-only tracking (BOT) or target motion analysis (TMA) which has been well studied in the literature [1] [7] [11]. On a single platform, the rate of change of the measurement must not be too small for the target to be observable. In addition, the platform must be able to outmaneuver the target. This means that the sensor platform must be moving with at least one degree of motion greater than the target [8]. For example, if the target is stationary, the sensor platform must be moving. If the target is moving at constant velocity, the sensor platform must be accelerating or performing a turn. It has been shown recently that a passive sensor can estimate the state of a target doing a coordinated turn without observer maneuver under a set of assumptions [9].

It is also possible to use a multiple passive sensor configuration to triangulate targets to provide better position estimates. The shortcoming to this approach is that it requires the sensors to communicate with each other (or to a fusion center) over a large baseline for good position estimation. It is costly to deploy such a sensor configuration over a large area without using radio communication. And if radio is used for communication, then the covert advantage can be lost. Therefore, there is great advantage to have passive sensors co-located on a single stationary platform and yet be able to initiate and track maneuvering targets.

The problem of target tracking in the presence of propagation delay has been studied recently. A number of approaches have been proposed, such as using a particle filter with a successive approximation approach (SAA) [13] [14] and the Unscented Gauss-Helmert Filter (UGHF) [18]. These approaches exploit the propagation delay to provide better estimates of the target state, and have better performance than a naive filter that ignores this phenomenon. The UGHF has been extended to use an interacting multiple model (IMM) estimator [6] [12] to track maneuvering targets [19]. However, these approaches still suffer from the same constraints as traditional BOT problems, i.e. the sensor platform must outmaneuver the target and the rate of change of measurement must not be too small.

A new approach, OOSM-AE, has been proposed in [17] to fuse measurements from two heterogeneous passive sensors on a single platform, one with negligible delay (such as EO or ESM sensor) and one with finite propagation delay (such as an acoustic sensor). The OOSM-AE handles the acoustic measurements as out-of-sequence measurements (OOSM) [2] [3] as they will arrive later than the EO measurements. The OOSM-AE has been demonstrated to improve observability and

TABLE I
List of Acronyms

Acronyms	Definition
AE	Acoustic and EO/ESM fusion
BOT	Bearing-only tracking
CT-H	Coordinated turn model with high process noise
CT-L	Coordinated turn model with low process noise
CV-L	Constant velocity model with low process noise
EO	Electro-optical (sensor)
ESM	Electronic support measures
IMM	Interacting multiple model (estimation)
IMMOOSM-AE	Extension of OOSM-AE that uses IMM
OOSM	Out-of-sequence measurement
OOSM-AE	OOSM algorithm which fuses acoustic and EO measurements
SAA	Successive approximation algorithm
TMA	Target motion analysis
UGHF	Unscented Gauss-Helmert filter

allows the sensor platform to be stationary as long as there is a sufficient rate of change in the measurements. This work enables greater operational flexibility as the platform no longer needs to be outmaneuvering the target in order to initiate and track the target. The main contribution of this paper is how to account properly for the time delay in one of the sensors and take advantage of this delay when tracking a maneuvering target using two sensors on a single non-maneuvering platform.

The aim of this paper is to extend the OOSM-AE fusion approach by combining it with interacting multiple model (IMM) estimation. The IMMOOSM-AE estimator captures target maneuvers by using additional motion models by calculating their likelihoods. Three motion models are used in the present work in IMMOOSM-AE: a low process noise nearly constant velocity model (CV-L), a high process noise coordinated turn model (CT-H) and a low process noise nearly coordinated turn model (CT-L).

Table I presents the lists of acronyms used in this paper.

Section II formulates the problem and defines the target state, measurement models and transition models. In Section III, the track initiation algorithm, which provides a starting track state and covariance for the filter, is described. In Section IV, the tracking filter is presented and the individual steps, such as IMM mixing, model based prediction, retrodiction and update, are described. In Section V, the two test scenarios (circular motion and connected S-turns) are presented and the results for IMMOOSM-AE and OOSM-AE (with three different levels of process noise) are provided. In Section VI, the conclusions are presented.

II. PROBLEM FORMULATION

Both the EO passive sensor, s_1 , and the acoustic passive sensor, s_2 , are assumed to be co-located on a stationary platform at $\mathbf{x}^s = [x^s, y^s]'$. For simplicity, $\dot{x}^s = \dot{y}^s = 0$ in this paper.

The target state for the EO sensor is

$$\mathbf{x}^E(t_k^{s_1}) = [x(t_k^{s_1}) \quad y(t_k^{s_1}) \quad \dot{x}(t_k^{s_1}) \quad \dot{y}(t_k^{s_1}) \quad \omega(t_k^{s_1})]'$$
 (1)

where $t_k^{s_1}$ is time at which the k^{th} EO signal is received by the EO sensor; x , y , \dot{x} , \dot{y} denote the position and velocity of the target and ω denotes the turn rate. We assume that the delay in propagation for EO signal is negligible, i.e. $t_k^{s_1} = t_k^{e_1}$, where $t_k^{e_1}$ is the time at which the signal is emitted from the target.

The measurement model for the EO sensor is¹

$$\mathbf{z}(t_k^{s_1}) = \tan^{-1} \left[\frac{x(t_k^{s_1}) - x^s}{y(t_k^{s_1}) - y^s} \right] + w_{s_1}(t_k^{s_1})$$
 (2)

where w_{s_1} is the zero mean white Gaussian measurement noise, with variance $\sigma_{s_1}^2$.

The transition model for the EO sensor is

$$\mathbf{x}^E(t_k^{s_1}) = f^*[\mathbf{x}^E(t_{k-1}^{s_1}, t_k^{s_1}, t_{k-1}^{s_1})] + v^*(t_k^{s_1}, t_{k-1}^{s_1})$$
 (3)

where $f(\cdot)$ is the transition function, v is the process noise, and $*$ stands for the different motion models given later in (30) and (33).

The acoustic sensor detects the target with a propagation delay. The target state for the acoustic sensor is

$$\mathbf{x}^A(t_j^{e_2}) = [x(t_j^{e_2}) \quad y(t_j^{e_2}) \quad \dot{x}(t_j^{e_2}) \quad \dot{y}(t_j^{e_2}) \quad \omega(t_j^{e_2}) \quad t_j^{e_2}]'$$
 (4)

where $t_j^{e_2}$ is time at which the j^{th} acoustic signal is emitted by the target. Note that the acoustic target state includes the emission time $t_j^{e_2}$.

The time delay the acoustic sensor detects the target state with is denoted by $\delta_{j,\ell}$. The relationship between the target acoustic emission time $t_j^{e_2}$ and the sensor receive time $t_\ell^{s_2}$ is

$$t_j^{e_2} = t_\ell^{s_2} - \delta_{j,\ell}$$
 (5)

where the delay is given by

$$\delta_{j,\ell} = \frac{r_{j,\ell}}{c^p}$$
 (6)

with $r_{j,\ell}$ the range from the target at $t_j^{e_2}$ to the sensor² at $t_\ell^{s_2}$, and c^p is the propagation speed of sound in the medium (air or water).

The measurement model for the acoustic sensor is

$$\mathbf{z}(t_j^{s_2}) = \tan^{-1} \left[\frac{x(t_j^{e_2}) - x^s}{y(t_j^{e_2}) - y^s} \right] + w_{s_2}(t_j^{e_2})$$
 (7)

where w_{s_2} is a zero-mean white Gaussian measurement noise with variance $\sigma_{s_2}^2$.

An illustration of the emission and reception times is given in Fig. 1.

Due to the propagation delay described in (5), the state transition model is implicit (see (8)) and a Gauss-Helmert model is required to represent the implicit state transition. The Gauss-Helmert model has been shown to be equivalent to the Markov model used in a Kalman

¹The present work assumes perfect data association.

²The sensor can move, but we considered a stationary sensor since this is the most difficult situation for passive tracking.

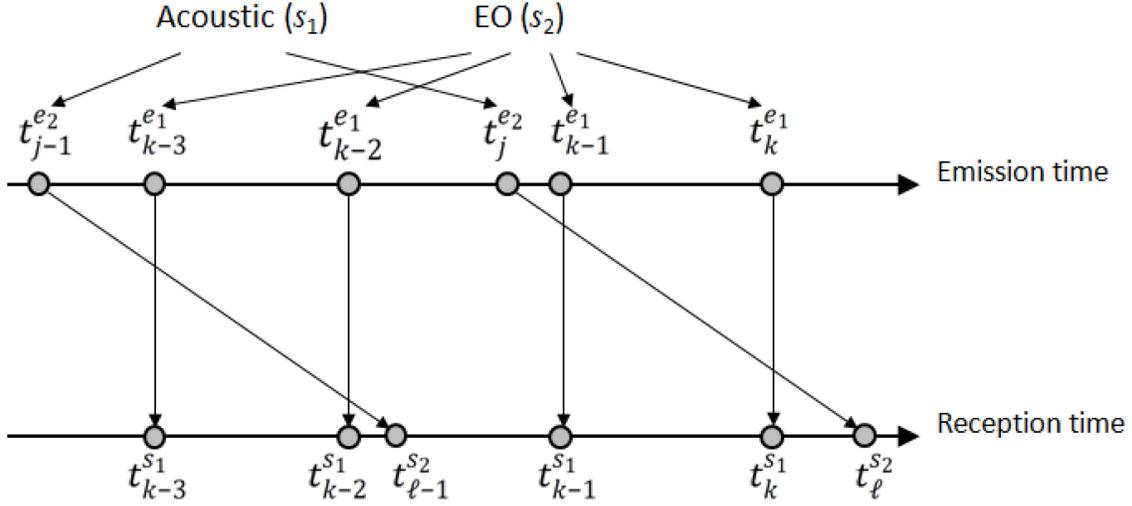


Fig. 1. Illustration of emission time and reception time

filter in [16]. However, there is no explicit formula obtainable for a retrodicted (or predicted) state from a current state. The Gauss-Helmert transition model is used for performing retrodiction and state update with the OOSM in Section IV-B.1 and the transition model is, instead of (3), of the following implicit form

$$g^*[\mathbf{x}^A(t_j^{e2}), \mathbf{x}^E(t_k^{s1})] + v^*(t_j^{e2}, t_k^{s1}) = \mathbf{0}_6 \quad (8)$$

where $\mathbf{x}^E(t_k^{s1})$ is the latest track state, $\mathbf{x}^A(t_j^{e2})$ is the state at the time at which the acoustic signal is emitted, $g(\cdot)$ is the Gauss-Helmert transition function, v is the process noise, $\mathbf{0}_6$ is the 6-dimensional zero vector and $*$ stands for the different motion models used in the IMMOOSM-AE given later in (55) and (69). Note that the \mathbf{x}^E state has dimension 5 while the \mathbf{x}^A state has dimension 6. The track is maintained in the 5-dimensional \mathbf{x}^E state while the 6-dimensional \mathbf{x}^A state is only used during retrodiction and OOSM innovation calculation using UGHF.

III. TRACK INITIATION

Given an initial batch of EO and acoustic bearing-only measurements from a single stationary platform,

$$\mathbf{z} = [z(t_1^s) \dots z(t_n^s)]' \quad s \in \{s_1, s_2\} \quad (9)$$

we want to initiate a track at time t_n^s . We define the initial track state, \mathbf{x} , at time t_n^s (the end of the initialization batch)

$$\mathbf{x} = [x(t_n^s) \quad y(t_n^s) \quad \dot{x}(t_n^s) \quad \dot{y}(t_n^s)]' \quad (10)$$

We assume that the target is moving at a constant velocity during the initialization batch.³ The relationship between \mathbf{x} and \mathbf{z} is

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{w} \quad (11)$$

³Other motion models can be used.

where component k of \mathbf{h} is

$$\mathbf{h}_k(\mathbf{x}, t_k^s) = \begin{cases} \tan^{-1} \left[\frac{x + \dot{x}(t_k^s - t_n^s) - x^s}{y + \dot{y}(t_k^s - t_n^s) - y^s} \right] & \text{if } s = s_1 \\ \tan^{-1} \left[\frac{x + \dot{x}(t_k^s - t_n^s - \delta_{j,k}) - x^s}{y + \dot{y}(t_k^s - t_n^s - \delta_{j,k}) - y^s} \right] & \text{if } s = s_2 \end{cases} \quad (12)$$

where $\delta_{j,k}$ is the time delay (6) for the acoustic signals and \mathbf{w} is the batch measurement noise.

We assume \mathbf{w} is zero mean Gaussian, with uncorrelated components. The covariance of \mathbf{w} (assuming for simplicity that $\sigma_{s_1} = \sigma_{s_2} = \sigma_b$) is

$$\mathbf{R} = \sigma_b^2 \mathbf{I}_n \quad (13)$$

where \mathbf{I}_n is the $n \times n$ identity matrix.

The estimate of the state \mathbf{x} can be obtained using the maximum likelihood (ML) approach by solving the following nonlinear least squares problem [4]

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \{[\mathbf{z} - \mathbf{h}(\mathbf{x})]' \mathbf{R}^{-1} [\mathbf{z} - \mathbf{h}(\mathbf{x})]\} \quad (14)$$

The Jacobian matrix of $\mathbf{h}(\cdot)$, required for solving the above,

$$\mathbf{H}(\mathbf{x}) = (\nabla_{\mathbf{x}} \mathbf{h}[\mathbf{x}]') \quad (15)$$

can be obtained by performing numerical partial differentiation on $\mathbf{h}(\cdot)$ with respect to each component of \mathbf{x} as in [17].

The track initiation algorithm is described in Table II where ℓ_{\max} is the maximum number of iterations before it terminates and $\mathbf{d}_{\text{threshold}}$ is the threshold value for the step size below which it terminates.

IV. THE TRACKING FILTER

The inputs to the dynamic state estimator are the EO and acoustic measurements. The EO measurements arrive instantaneously, while the acoustic measurements arrive with a propagation delay. The estimator updates the state at the EO measurement times and treats the acoustic measurements as out-of-sequence measure-

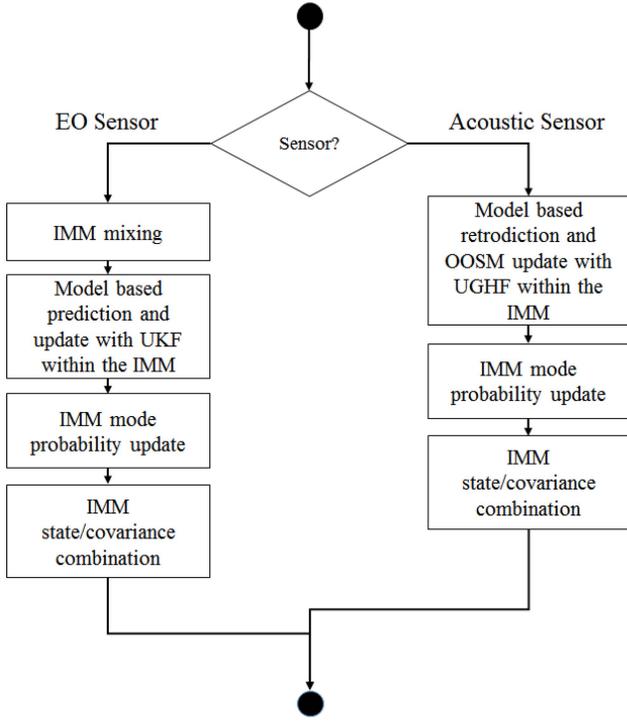


Fig. 2. Overview of IMMOOSM-AE estimator

$$\Pi(T) = \frac{1}{\lambda} \begin{bmatrix} \lambda_2 + \lambda_3 + \lambda_1 e^{-\lambda T} & \alpha[\lambda_1 - \lambda_1 e^{-\lambda T}] & (1 - \alpha)[\lambda_1 - \lambda_1 e^{-\lambda T}] \\ \beta[\lambda_2 - \lambda_2 e^{-\lambda T}] & \lambda_1 + \lambda_3 + \lambda_2 e^{-\lambda T} & (1 - \beta)[\lambda_2 - \lambda_2 e^{-\lambda T}] \\ \gamma[\lambda_3 - \lambda_3 e^{-\lambda T}] & (1 - \gamma)[\lambda_3 - \lambda_3 e^{-\lambda T}] & \lambda_1 + \lambda_2 + \lambda_3 e^{-\lambda T} \end{bmatrix} \quad (19)$$

ments (OOSM). An overview of a single cycle of the estimator is shown in Fig. 2.

The IMM incorporates 3 motion models: CV-L (nearly constant velocity, with low process noise—white noise acceleration), CT-H and CT-L (nearly coordinated turn with high and low noise). Model 1 is CV-L, model 2 is CT-H and model 3 is CT-L. The purpose of the CV-L is to capture the constant velocity motion. Likewise, the purpose of the CT-L is to capture the ongoing coordinated turn maneuvers. The purpose of the CT-H model is to facilitate the abrupt change from CV-L to CT-L and vice versa (turn onset and termination). The turn rate, ω , must be allowed to switch from zero (during constant velocity) to a non-zero value (during coordinated turn) in a short time. Using only CV-L and CT-L with low process noise in the turn rate will not enable the estimator to follow this change quickly.

A. EO Measurements

This section describes how the EO measurements are handled in the IMM estimator.

1) IMM Mixing:

In IMM estimation, a mode probability, μ_k^i , is calculated for each motion model i at time k . The IMM

TABLE II
Track Initiation

- 1) Compute initial estimate $\hat{\mathbf{x}}^0$ based on EO bearing measurement at t_n^s and a moderate range and zero velocity.
- 2) Initialize $\ell = 0$.
- 3) While $\ell < \ell_{\max}$ or $|\mathbf{d}| > \mathbf{d}_{\text{threshold}}$
 - a) $\mathbf{P}^\ell = [\mathbf{H}(\hat{\mathbf{x}}^\ell)'\mathbf{R}^{-1}\mathbf{H}(\hat{\mathbf{x}}^\ell)]^{-1}$ (16)
 - b) $\mathbf{d} = \mathbf{P}^\ell[\mathbf{H}(\hat{\mathbf{x}}^\ell)'\mathbf{R}^{-1}[\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}^\ell)]]$ (17)
 - c) $\hat{\mathbf{x}}^{\ell+1} = \hat{\mathbf{x}}^\ell + \mathbf{d}$ (18)
- 4) Assign $\hat{\mathbf{x}}^\ell$ and \mathbf{P}^ℓ as the initiated track state and covariance.

estimation is able to capture the motion model change by dynamically adjusting μ^i according to the filter update. For example, a target can move with nearly constant velocity and subsequently perform a coordinated turn. Then, μ^1 for CV-L model will become the highest among the different motion models during its constant velocity motion at the start and, subsequently, μ^3 for the CT-L model will become highest during the turning motion. The evolution of the mode probabilities, μ^i , depends on the transition probability matrix Π .

The transition probability matrix, Π , used in the present work is a generalization of the discretized continuous-time Markov chain transition probability matrix from Eq. (2.6.6-15) in [5].

where, with λ_m , $m = 1, 2, 3$, the transition probability rates (their inverses are the expected sojourn times in the corresponding states of the Markov chain),

$$\lambda = \sum_{n=1}^3 \lambda_n \quad (20)$$

$$T = |t_k^{s_1} - t_{k-1}^{s_1}| \quad (21)$$

for EO IMM mixing (the prediction time interval) or

$$T = |t_j^{e_2} - t_k^{s_1}| \quad (22)$$

for acoustic OOSM mode probability update,⁴ and α , β and γ are normalizing factors which are introduced to keep the sum of the row elements of Π to be unity.⁵

⁴This is the difference between the time stamp of the acoustic measurement and the time for which the state update is performed (the retrodiction interval, see (39) in the sequel). See Fig. 1 for the time notations.

⁵The 2-dimensional Markov chain transition matrix is rigorously derived in Papoulis [15] from the continuous-time chain with appropriate transition rates. The 3-dimensional continuous-time Markov chain does not have an explicit transition matrix, so this is the generalization of the 2-dimensional Markov chain transition matrix to 3 dimensions by adding another transition rate.

In IMMOOSM-AE, Π is a 3×3 matrix, since three motion models are used: CV-L, CT-H, and CT-L. The CV model assumes the turn rate to be zero while the two CT models include the turn rate. Thus, unbiased mixing must be done [20]. The weighted sum of the probabilities corresponding to turning (μ^2, μ^3) from the two CT models are transferred to the modified state and covariance for the CV model for the purpose of mixing. This ensures that the resulting elements corresponding to turn rate ($\hat{\omega}^2, \hat{\omega}^3$) in the two CT models are unbiased after mixing. The modification is done according to [20] as follows:

$$\hat{\mathbf{x}}_M^{\text{E1}} = \begin{bmatrix} \hat{\mathbf{x}}_c^{\text{E1}} \\ \hat{\omega}^2 \mu_{2|1} + \hat{\omega}^3 \mu_{3|1} \end{bmatrix} \quad (23)$$

$$\mathbf{P}_M^{\text{E1}} = \begin{bmatrix} \mathbf{P}_c^{\text{E1}} & 0 \\ 0 & \mathbf{P}_\omega^{\text{E2}} \mu_{2|1} + \mathbf{P}_\omega^{\text{E3}} \mu_{3|1} \end{bmatrix} \quad (24)$$

where $\hat{\mathbf{x}}_c^{\text{E1}}$ and \mathbf{P}_c^{E1} are the blocks common to both CV and CT models, i.e. corresponding to the x , y , \dot{x} and \dot{y} states and $\mu_{2|1}$, $\mu_{3|1}$ are the IMM mixing probabilities [4].

TABLE III
IMM Mixing

$$\mu^i(t_k^{s_1} | t_{k-1}^{s_1}) = \sum_{n=1}^m \Pi_{ni} \mu^n(t_{k-1}^{s_1}) \quad (25)$$

$$\mu^{n|i}(t_{k-1}^{s_1}) = \frac{\Pi_{ni} \mu^n(t_{k-1}^{s_1})}{\mu^i(t_k^{s_1} | t_{k-1}^{s_1})} \quad (26)$$

$$\hat{\mathbf{x}}^{\text{E0}i}(t_{k-1}^{s_1}) = \sum_{n=1}^m \hat{\mathbf{x}}^{\text{E}n}(t_{k-1}^{s_1}) \mu^{n|i}(t_{k-1}^{s_1}) \quad (27)$$

$$\mathbf{P}^{\text{E0}i}(t_{k-1}^{s_1}) = \sum_{n=1}^m \mu^{n|i}(t_{k-1}^{s_1}) [\mathbf{P}^{\text{E}n}(t_{k-1}^{s_1}) + (\hat{\mathbf{x}}^{\text{E}n}(t_{k-1}^{s_1}) - \hat{\mathbf{x}}^{\text{E0}i}(t_{k-1}^{s_1})) (\hat{\mathbf{x}}^{\text{E}n}(t_{k-1}^{s_1}) - \hat{\mathbf{x}}^{\text{E0}i}(t_{k-1}^{s_1}))'] \quad (28)$$

$$(\hat{\mathbf{x}}^{\text{E}n}(t_{k-1}^{s_1}) - \hat{\mathbf{x}}^{\text{E0}i}(t_{k-1}^{s_1}))'] \quad (29)$$

The IMM mixing uses the mixing probability (26) based on the transition probability matrix, Π , and com-

putes the initial estimate $\hat{\mathbf{x}}_{k-1}^{\text{E0}i}$ and covariance $\mathbf{P}_{k-1}^{\text{E0}i}$ according to (27) and (28), respectively, where i corresponds to each model. The IMM mixing steps are given in Table. III, and the mixed estimates and covariances from t_{k-1} are used as initial condition for the mode-matched filters at time t_k in Section IV-A.2.

2) Prediction and Update using UKF:

The unscented Kalman filter (UKF) is used to predict and update the state for each mode with the EO measurements.

The transition model, f^{CV} , and process noise, \mathbf{Q}^{CV} , for the CV model⁶ are given below.

$$f^{\text{CV}}[\mathbf{x}^{\text{E}}(t_{k-1}^{s_1}), T_{k,k-1}] = \begin{bmatrix} 1 & 0 & T_{k,k-1} & 0 & 0 \\ 0 & 1 & 0 & T_{k,k-1} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}^{\text{E}}(t_{k-1}^{s_1}) \quad (30)$$

$$E[v^{\text{CV}}(\cdot)v^{\text{CV}}(\cdot)'] = \mathbf{Q}^{\text{CV}}(t_k^{s_1} - t_{k-1}^{s_1}) = \begin{bmatrix} \frac{T_{k,k-1}^3}{3} & 0 & \frac{T_{k,k-1}^2}{2} & 0 & 0 \\ 0 & \frac{T_{k,k-1}^3}{3} & 0 & \frac{T_{k,k-1}^2}{2} & 0 \\ \frac{T_{k,k-1}^2}{2} & 0 & T_{k,k-1} & 0 & 0 \\ 0 & \frac{T_{k,k-1}^2}{2} & 0 & T_{k,k-1} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} q \quad (31)$$

where

$$T_{k,k-1} = t_k^{s_1} - t_{k-1}^{s_1} \quad (32)$$

where q is the process noise power spectral density (PSD) that affects the x , y , \dot{x} and \dot{y} states. The physical dimension of q is acceleration²/frequency.

The transition model, f^{CT} , and process noise, \mathbf{Q}^{CT} , for the CT model are given below.

$$f^{\text{CT}}[\mathbf{x}^{\text{E}}(t_{k-1}^{s_1}), T_{k,k-1}] = \begin{bmatrix} 1 & 0 & \frac{\sin[\omega(t_{k-1}^{s_1})T_{k,k-1}]}{\omega(t_{k-1}^{s_1})} & -\frac{1 - \cos[\omega(t_{k-1}^{s_1})T_{k,k-1}]}{\omega(t_{k-1}^{s_1})} & 0 \\ 0 & 1 & \frac{1 - \cos[\omega(t_{k-1}^{s_1})T_{k,k-1}]}{\omega(t_{k-1}^{s_1})} & \frac{\sin[\omega(t_{k-1}^{s_1})T_{k,k-1}]}{\omega(t_{k-1}^{s_1})} & 0 \\ 0 & 0 & \cos[\omega(t_{k-1}^{s_1})T_{k,k-1}] & -\sin[\omega(t_{k-1}^{s_1})T_{k,k-1}] & 0 \\ 0 & 0 & \sin[\omega(t_{k-1}^{s_1})T_{k,k-1}] & \cos[\omega(t_{k-1}^{s_1})T_{k,k-1}] & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}^{\text{E}}(t_{k-1}^{s_1}) \quad (33)$$

⁶This is actually a white noise acceleration (WNA) model.

TABLE IV
The UKF steps

- 1) Generate the preliminary sigma points and weights based on the initial estimate $\hat{\mathbf{x}}^{E0i}(t_{k-1}^{s1})$ and covariance $\mathbf{P}^{E0i}(t_{k-1}^{s1})$ [10]; i denotes the mode.
- 2) Predict the preliminary sigma points using transition model, f .
- 3) Compute predicted $\hat{\mathbf{x}}^{Ei}(t_k^{s1} | t_{k-1}^{s1})$ and $\mathbf{P}^{Ei}(t_k^{s1} | t_{k-1}^{s1})$ based on propagated sigma points and weights.
- 4) Add the model process noise \mathbf{Q}^* to $\mathbf{P}^{Ei}(t_k^{s1} | t_{k-1}^{s1})$ where \mathbf{Q}^{CV} and \mathbf{Q}^{CT} are given in (31) and (34).
- 5) Recalculate the sigma points to account for the added process noise covariance.
- 6) Compute the predicted $\hat{\mathbf{z}}^i(t_k^{s1} | t_{k-1}^{s1})$ based on the propagated sigma points and weight using the measurement model.
- 7) Calculate the innovation covariance $S^{Ei}(t_k^{s1})$.
- 8) Use the sensor measurement $\mathbf{z}(t_k^{s1})$ to obtain the innovation $\nu^{Ei}(t_k^{s1})$, the updated state $\hat{\mathbf{x}}^{Ei}(t_k^{s1})$, and covariance $\mathbf{P}^{Ei}(t_k^{s1})$.

$$\begin{aligned}
& E[\nu^{CT}(\cdot)\nu^{CT}(\cdot)'] \\
&= \mathbf{Q}^{CT}(t_k^{s1} - t_{k-1}^{s1}) \\
&= \mathbf{Q}^{CV}(t_k^{s1} - t_{k-1}^{s1}) + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & T_{k,k-1} \end{bmatrix} q_\omega
\end{aligned} \tag{34}$$

where

$$T_{k,k-1} = t_k^{s1} - t_{k-1}^{s1} \tag{35}$$

where q_ω is the process noise PSD that affects the ω state (angular rate). The physical dimension of q_ω is (angular acceleration)²/frequency

The UKF prediction and update are performed for each model as described in Table IV.

- 3) Mode Probability Update and State Estimate/
Covariance Combination:

With the mode-conditioned innovation, $\nu^{Ei}(t_k^{s1})$, and its covariance, $S^{Ei}(t_k^{s1})$, the mode probabilities, $\mu^i(t_k^{s1})$, are updated as

$$\mu^i(t_k^{s1}) = \frac{\mu^i(t_k^{s1} | t_{k-1}^{s1}) \mathcal{N}(\nu^{Ei}(t_k^{s1}); 0, S^{Ei}(t_k^{s1}))}{\sum_{n=1}^m \mu^n(t_k^{s1} | t_{k-1}^{s1}) \mathcal{N}(\nu^{En}(t_k^{s1}); 0, S^{En}(t_k^{s1}))} \tag{36}$$

where $\mathcal{N}(\nu^{Ei}(t_k^{s1}); 0, S^{Ei}(t_k^{s1}))$ is the model likelihood function based on the latest measurement.

The combined state estimate, $\hat{\mathbf{x}}^E(t_k^{s1})$, and covariance, $\mathbf{P}^E(t_k^{s1})$, are obtained as

$$\hat{\mathbf{x}}^E(t_k^{s1}) = \sum_{n=1}^m \mu^n(t_k^{s1}) \hat{\mathbf{x}}^{En}(t_k^{s1}) \tag{37}$$

$$\begin{aligned}
\mathbf{P}^E(t_k^{s1}) &= \sum_{n=1}^m \mu^n(t_k^{s1}) [\mathbf{P}^{En}(t_k^{s1}) \\
&+ (\hat{\mathbf{x}}^{En}(t_k^{s1}) - \hat{\mathbf{x}}^E(t_k^{s1})) (\hat{\mathbf{x}}^{En}(t_k^{s1}) - \hat{\mathbf{x}}^E(t_k^{s1}))']
\end{aligned} \tag{38}$$

B. Acoustic Measurements

This section describes how the acoustic measurements are handled in the IMM estimator. Acoustic measurements arrive later than the EO measurement due to the slower propagation of sound, i.e. they are OOSM and they are incorporated into the IMM according to the procedure described in Sec. 2.6.6 of [5]. The current state estimate must be retrodicted back in time to the time of acoustic signal emission before the update. This is handled by the Unscented Gauss-Helmert Filter (UGHF) retrodiction and the update with the OOSM is then carried out as described in the sequel.

- 1) Retrodiction and OOSM innovation calculation using UGHF:

The process noise is not taken into consideration in the state for algorithm C⁷ (see Sec 2.6.3 of [5]).⁸ The Gauss-Helmert model (GHM) is thus replaced by

$$g^*[\hat{\mathbf{x}}^A(t_j^{e2} | t_k^{s1}), \hat{\mathbf{x}}^E(t_k^{s1})] = \mathbf{0}_6 \tag{39}$$

where $\hat{\mathbf{x}}^A(t_j^{e2} | t_k^{s1})$ is the (6-dimensional) retrodicted state, which also includes the emission time, required for the measurement update and $\hat{\mathbf{x}}^E(t_k^{s1})$ is the latest (5-dimensional) track state estimate.

The Markov model used in a Kalman filter relies on an explicit form of the state transition model. In contrast, the GHM is for situations where there is only an implicit transition model and it uses the Gauss-Newton algorithm to obtain the retrodicted state by solving (39). The Gauss-Newton iteration with index p is

$$\begin{aligned}
& [\hat{\mathbf{x}}^A(t_j^{e2} | t_k^{s1})]^p \\
&= [\hat{\mathbf{x}}^A(t_j^{e2} | t_k^{s1})]^{p-1} - \mathbf{A}^{-1} g^* [[\hat{\mathbf{x}}^A(t_j^{e2} | t_k^{s1})]^{p-1}, \hat{\mathbf{x}}^E(t_k^{s1})]
\end{aligned} \tag{40}$$

where \mathbf{A} is the Jacobian matrix

$$\mathbf{A} = \frac{\partial g^* [[\hat{\mathbf{x}}^A(t_j^{e2} | t_k^{s1})]^p, \hat{\mathbf{x}}^E(t_k^{s1})]}{\partial [[\hat{\mathbf{x}}^A(t_j^{e2} | t_k^{s1})]^p]} \tag{41}$$

and $[\cdot]^p$ indicates the estimated value in the p th iteration. The algorithm is terminated when $p = 1000$ or

$$\frac{|[\hat{\mathbf{x}}^A(t_j^{e2} | t_k^{s1})]^p - [\hat{\mathbf{x}}^A(t_j^{e2} | t_k^{s1})]^{p-1}|}{|[\hat{\mathbf{x}}^A(t_j^{e2} | t_k^{s1})]^{p-1}|} < 0.1 \tag{42}$$

The GHM for CV and CT can be found in Appendix A and Appendix B respectively. The starting point for the Gauss-Newton algorithm, $[\hat{\mathbf{x}}^A(t_j^{e2} | t_k^{s1})]^0$, is computed by assuming the initial emission time

$$[t_j^{e2}]^0 = t_k^{s1} - \frac{\sqrt{x(t_k^{s1})^2 + y(t_k^{s1})^2}}{c^p} \tag{43}$$

⁷This is the simplest retrodiction algorithm, which does not take into account the process noise.

⁸This is one of the algorithms presented in [5], chosen for the present work.

TABLE V
Retrodiction and Update with OOSM in UGHF

- 1) Generate sigma points and weights based on $\hat{\mathbf{x}}^{Ei}(t_k^{s1})$ and $\mathbf{P}^{Ei}(t_k^{s1})$; i denotes the mode.
- 2) Retrodicit sigma points with the transition model, g , using the Gauss-Newton algorithm.
- 3) Compute $\hat{\mathbf{x}}^{Ai}(t_j^{e2} | t_k^{s1})$ and $\mathbf{P}^{Ai}(t_j^{e2} | t_k^{s1})$ based on propagated sigma points and weights.
- 4) Compute the retrodicted measurement $\hat{\mathbf{z}}^i(t_j^{e2} | t_k^{s1})$ based on the propagated sigma points and weights using the measurement model.
- 5) Calculate innovation covariance $S^{Ai}(t_k^{s1}, t_j^{e2})$.
- 6) Use the sensor measurement $\mathbf{z}(t_j^{s2})$ to obtain innovation $\nu^{Ai}(t_k^{s1}, t_j^{e2})$, the updated state $\hat{\mathbf{x}}^{Ei}(t_k^{s1}, t_j^{e2})$ and the updated covariance $\mathbf{P}^{Ei}(t_k^{s1}, t_j^{e2})$.

With the initial emission time $[t_j^{e2}]^0$, the remaining elements in the initial $[\hat{\mathbf{x}}^A(t_j^{e2} | t_k^{s1})]^0$ can be computed from $\hat{\mathbf{x}}^E(t_k^{s1})$ using the standard CV and CT transition model, f , given in (30) and (33), respectively.

The UGHF retrodiction and its update with the OOSM are done for each model as described in Table V.

- 2) Mode probability and state update with the OOSM and state/covariance combination:

The mode probability update with the acoustic measurements is described here. The transition probability matrix $\Pi(T)$ from (19), with $T_{j,k} = |t_j^{e2} - t_k^{s1}|$, is used for the mode probability update with the OOSM as

$$\begin{aligned} \mu^i(t_k^{s1}, t_j^{e2}) \\ = \frac{1}{c} \left[\sum_{n=1}^m \mathcal{N}(\nu^{An}(t_k^{s1}, t_j^{e2}); 0, S^{An}(t_k^{s1}, t_j^{e2})) \Pi_{in} \right] \mu^i(t_k^{s1}) \end{aligned} \quad (44)$$

where $\mu^i(t_k^{s1}, t_j^{e2})$ is the updated mode probability at t_k^{s1} using the OOSM from t_j^{e2} .

$$c = \sum_{\ell=1}^m \sum_{n=1}^m \mathcal{N}(\nu^{An}(t_k^{s1}, t_j^{e2}); 0, S^{An}(t_k^{s1}, t_j^{e2})) \Pi_{ln} \mu^\ell(t_k^{s1}) \quad (45)$$

The state and covariance combination is done following the update as follows:

$$\hat{\mathbf{x}}^E(t_k^{s1}, t_j^{e2}) = \sum_{n=1}^m \mu^n(t_k^{s1}, t_j^{e2}) \hat{\mathbf{x}}^{En}(t_k^{s1}, t_j^{e2}) \quad (46)$$

$$\begin{aligned} \mathbf{P}^E(t_k^{s1}, t_j^{e2}) = \sum_{n=1}^m \mu^n(t_k^{s1}, t_j^{e2}) [\mathbf{P}^{En}(t_k^{s1}, t_j^{e2}) \\ + (\hat{\mathbf{x}}^{En}(t_k^{s1}, t_j^{e2}) - \hat{\mathbf{x}}^E(t_k^{s1}, t_j^{e2})) \\ (\hat{\mathbf{x}}^{En}(t_k^{s1}, t_j^{e2}) - \hat{\mathbf{x}}^E(t_k^{s1}, t_j^{e2}))'] \end{aligned} \quad (47)$$

V. SCENARIOS AND RESULTS

Two scenarios are generated and tested by IMMOOSM-AE and OOSM-AE. Three variants of OOSM-AE with different levels of process noise are

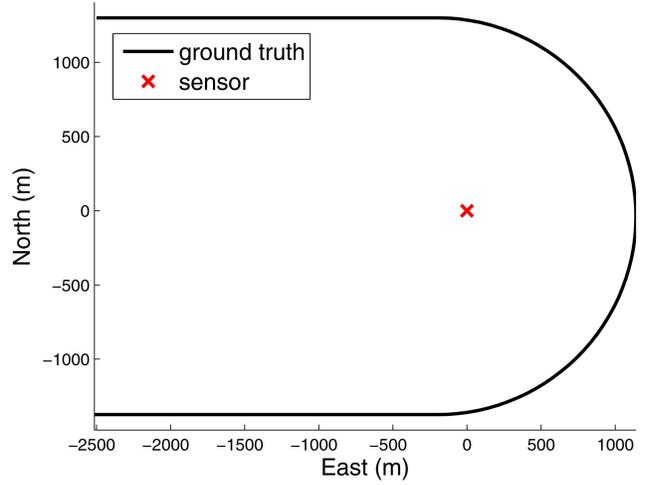


Fig. 3. U-turn scenario

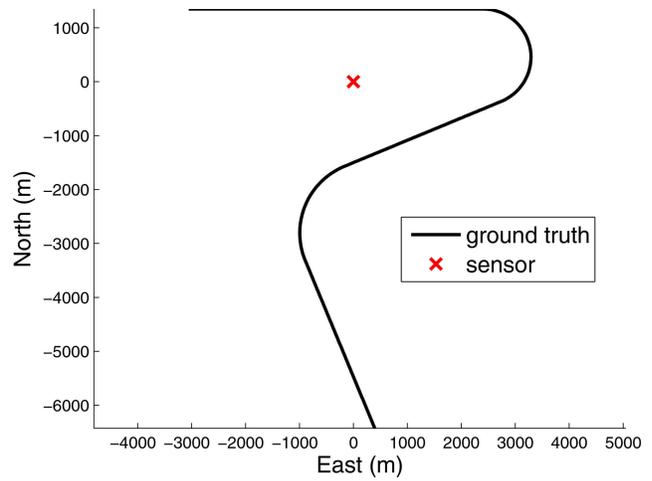


Fig. 4. S-turn scenario

tested. In the U-turn scenario, the target starts at $(-2500, 1300)$ and travels east at 70 m/s for 35 s. Then, it makes a right 3 deg/s turn for 60 s (acceleration of 3.7 m/s²). Finally, it travels west at 70 m/s for 35 s. An illustration is given in Fig. 3. In the S-turn scenario, the target starts at $(-3000, 1350)$ and travels east at 70 m/s for 80 s. Next, it makes a right 4.5 deg/s turn for 35 s (acceleration of 5.5 m/s²). Next, it travels west-southwest at 70 m/s for 45 s. Next, it makes a left 3 deg/s turn for 30 s (acceleration of 3.7 m/s²). Finally, it travels south-south-east at 70 m/s for 50 s. This is illustrated in Fig. 4.

For OOSM-AE which consists of a single CV model using Eq. (1.5.2-5) from [5], the process noise PSD for the single-model (compromise) filter is chosen as

$$q = a_{ave}^2 \tau \quad (48)$$

with $a_{ave} = 1.4$ m/s² (compromise between exact CV motion and turn which has $a = 3.5$ m/s²) and $\tau = 1$ s, one obtains $q = 2$ m²/s³; τ is defined as the time interval

over which the acceleration is assumed to be approximately constant. The scenarios are tested by three variants of OOSM-AE: OOSM-AE-Q1 with $q_1 = 2 \text{ m}^2/\text{s}^3$, OOSM-AE-Q2 with $q_2 = 4 \text{ m}^2/\text{s}^3$ (which corresponds to $a = 2 \text{ m/s}^2$) and OOSM-AE-Q3 with $q_3 = 9 \text{ m}^2/\text{s}^3$ (which corresponds to $a = 3 \text{ m/s}^2$).

IMMOOSM-AE consists of 3 models: CV-L, CT-H and CT-L. For the CT-H and the CT-L model, the process noise PSD is obtained using the formula (1.5.3-5) from [5] (modified for turn rate increments)

$$q_\omega = \left(\frac{\Delta\omega}{\tau} \right)^2 \tau = \frac{(\Delta\omega)^2}{\tau} \quad (49)$$

With $\Delta\omega = 1 \text{ deg/s}$ and $\tau = 1 \text{ s}$, one obtains $q_\omega^H = 1 \text{ deg}^2/\text{s}^3$ for the CT-H model. With $\Delta\omega = 0.1 \text{ deg/s}$ and $\tau = 1 \text{ s}$, one has $q_\omega^L = 0.01 \text{ deg}^2/\text{s}^3$ for the CT-L model. For the CV-L model, the process noise PSD is chosen using acceleration $a = \Delta\omega v$ where $v = 70 \text{ m/s}$ and $\Delta\omega = 0.1 \text{ deg/s}$. This yields

$$a = \Delta\omega v = 0.1 \text{ deg/s} \cdot 70 \text{ m/s} = \frac{0.1\pi}{180} \cdot 70 = 0.12 \text{ m/s}^2 \quad (50)$$

Using (48) and (50), the PSD for CV-L is taken as $q = 1.5 \cdot 10^{-2} \text{ m}^2/\text{s}^3$.

The exponential sojourn time distribution parameter for the computing the transition probability are set to be $\lambda_1 = \lambda_3 = 10^{-2} \text{ s}^{-1}$ and $\lambda_2 = 0.2 \text{ s}^{-1}$. The normalizing factors for the transition probability matrix (19) are set as follows: $\alpha = 0.9$, $\beta = 0.5$ and $\gamma = 0.1$.

For both of the scenarios, the sensor is stationary at the origin. Both the EO and acoustic sensor have measurement error with $\sigma_b = 1 \text{ deg}$. The sampling period for the EO sensor is 1 s, while the sampling period for the acoustic sensor is 2 s. The propagation speed of sound, c^p , is 344 m/s.

The first 20 s of measurements data are used for track initiation assuming exact CV motion. Subsequently, OOSM-AE-Q1, OOSM-AE-Q2, OOSM-AE-Q3 and IMMOOSM-AE are used to track the target.

100 Monte Carlo runs are generated and the average root-mean-square error (RMSE) for position and velocity are presented in Tables VI and VII. The average position $\overline{\text{RMSE}}_k^p(N)$ at time k from N Monte Carlo runs, is calculated as follows

$$\overline{\text{RMSE}}_k^p(N) = \sqrt{\frac{1}{N} \sum_{n=1}^N \|\hat{\mathbf{x}}_k^p(n) - \mathbf{x}_k^{\text{pg}}\|^2} \quad (51)$$

where $\hat{\mathbf{x}}_k^p(n)$ is the position state estimate at time k for run n , \mathbf{x}_k^{pg} is the position ground truth⁹ at time k . The velocity $\overline{\text{RMSE}}_k^v(N)$ is calculated in the same manner, by replacing $\hat{\mathbf{x}}_k^p(n)$ and \mathbf{x}_k^{pg} with the velocity state estimate $\hat{\mathbf{x}}_k^v(n)$ and velocity ground truth \mathbf{x}_k^{vg} respectively.

⁹The ground truth is not noisy in the example considered, it only exhibits maneuvers that have to be modeled as process noise by the tracker. It should be pointed out that white process noise is needed for the state to be a Markov process in order to estimate it recursively.

TABLE VI
Average RMSE^p (in m) for each scenario and 95% confidence region for the true RMSE^p

	U-turn scenario	S-turn scenario
OOSM-AE-Q1	217.8 [191.3, 252.7]	539.2 [473.6, 625.5]
OOSM-AE-Q2	208.1 [182.8, 241.4]	476.0 [418.1, 552.2]
OOSM-AE-Q3	182.8 [160.6, 212.1]	416.6 [365.9, 483.3]
IMMOOSM-AE	84.6 [74.3, 98.1]	188.3 [165.4, 218.4]

TABLE VII
Average RMSE^v (in m/s) for each scenario and 95% confidence region for the true RMSE^v

	U-turn scenario	S-turn scenario
OOSM-AE-Q1	30.6 [26.9, 35.5]	30.8 [27.1, 35.7]
OOSM-AE-Q2	28.7 [25.2, 33.3]	28.6 [25.1, 33.2]
OOSM-AE-Q3	26.3 [23.1, 30.5]	27.0 [23.7, 31.3]
IMMOOSM-AE	11.3 [9.9, 13.1]	11.9 [10.5, 13.8]

TABLE VIII
Track loss for the S-turn scenario

	No. of lost tracks
OOSM-AE-Q1	87
OOSM-AE-Q2	76
OOSM-AE-Q3	57
IMMOOSM-AE	1

The 95% confidence region for the true position RMSE given the average $\overline{\text{RMSE}}(N)$ is, according to Appendix C, given by the following interval.

$$\frac{\text{RMSE}}{\overline{\text{RMSE}}(N)}$$

$$\in \left[\left(\frac{1}{N} \chi_N^2(97.5\%) \right)^{-1/2}, \left(\frac{1}{N} \chi_N^2(2.5\%) \right)^{-1/2} \right] \quad (52)$$

which for $N = 100$ becomes

$$\text{RMSE} \in [0.88\overline{\text{RMSE}}(N), 1.16\overline{\text{RMSE}}(N)] \quad (53)$$

The above is used for the average over all time steps, i.e.,

$$\overline{\text{RMSE}}^p(N) = \frac{\sum_{k=1}^K \overline{\text{RMSE}}_k^p(N)}{K} \quad (54)$$

with K being the total number of time steps. The 95% confidence region for the average velocity RMSE is obtained in the same manner. These 95% confidence regions are presented in Tables VI and VII for position and velocity, respectively.

Clearly, the IMM estimation shows its value versus any single-model based filter, no matter what the latter's choice of process noise PSD.

The track is defined to be lost when the distance between the track position estimate and the ground truth becomes greater than 1500 m for more than 10 s consecutively. There are no lost tracks for the U-turn scenario for all four trackers. The number of lost tracks for the S-turn scenario are presented in Table VIII. It can

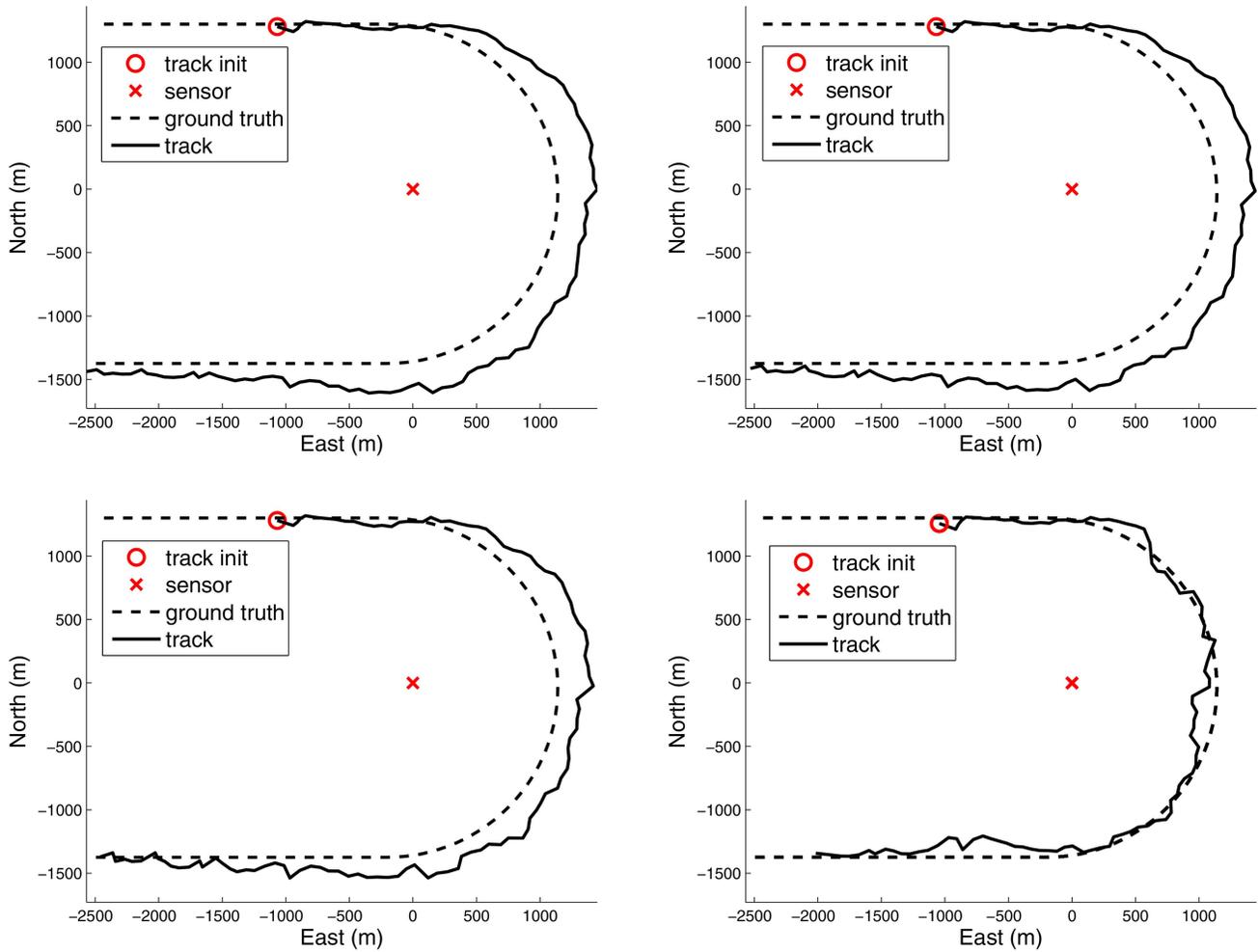


Fig. 5. OOSM-AE-Q1 (top left), OOSM-AE-Q2 (top right), OOSM-AE-Q3 (bottom left) and IMMOOSM-AE (bottom right) single run result for the U-turn scenario

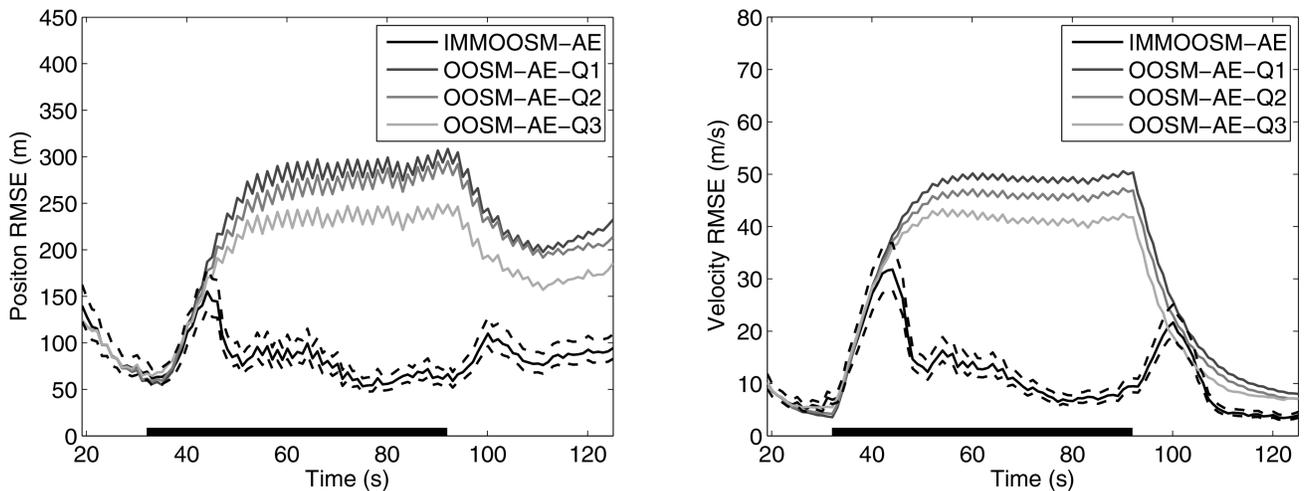


Fig. 6. Average position RMSE (left) and average velocity RMSE (right) for the U-turn scenario. The dashed lines around the IMMOOSM-AE curve represent the variability (2σ) of its performance.

be observed that the IMM filter is able to track the target without very little lost tracks, while the single-models filter suffer from significant lost tracks. The number of lost tracks is observed to be higher for the lower process noise single-model filter.

For a single run of the U-turn scenario, the estimated trajectories with OOSM-AE-Q1, OOSM-AE-Q2, OOSM-AE-Q3 and IMMOOSM-AE are shown in Fig 5. Again, no single-model filter performs even close to the IMM. It can be observed that the higher process noise

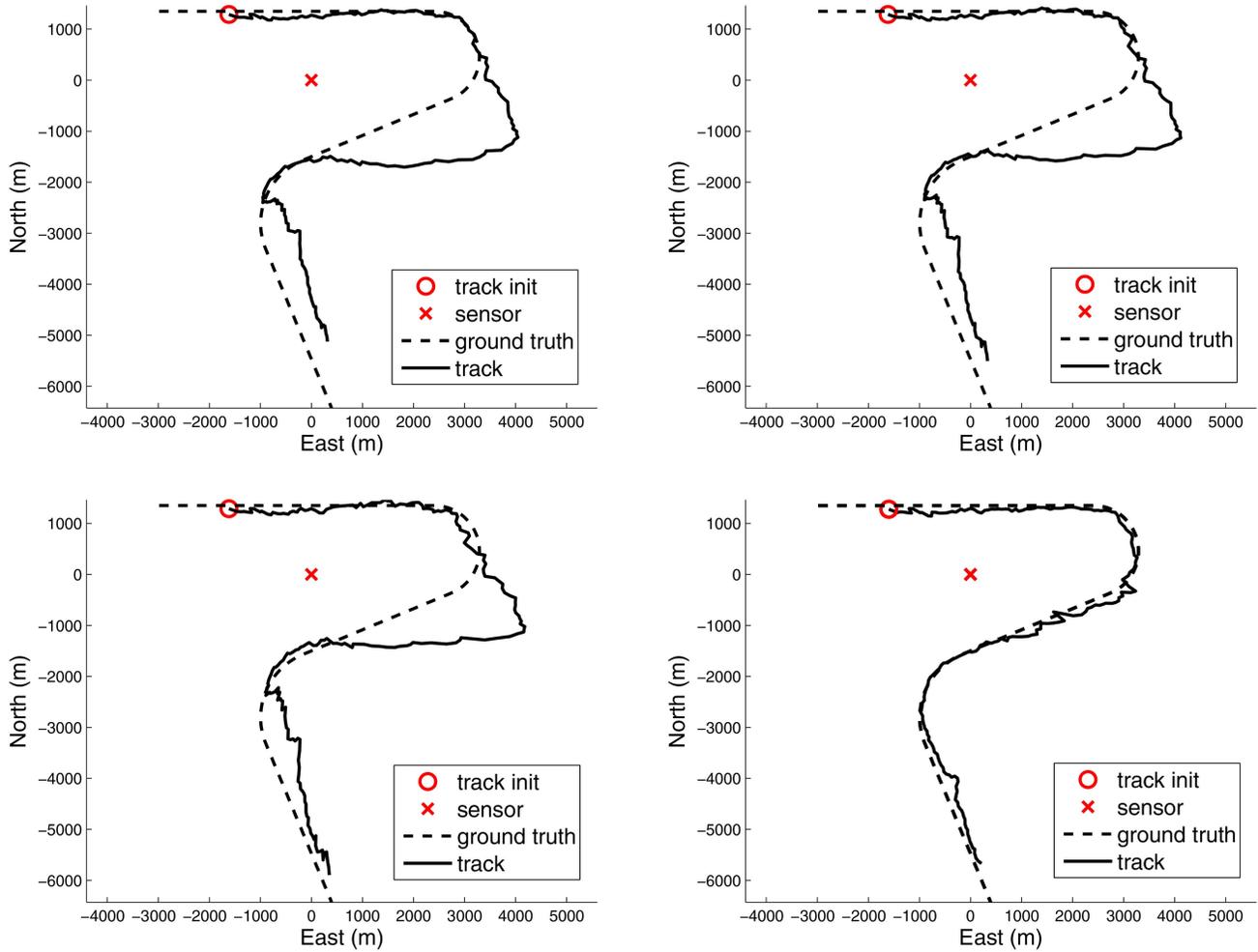


Fig. 7. OOSM-AE-Q1 (top left), OOSM-AE-Q2 (top right), OOSM-AE-Q3 (bottom left) and IMMOOSM-AE (bottom right) single run result for the S-turn scenario

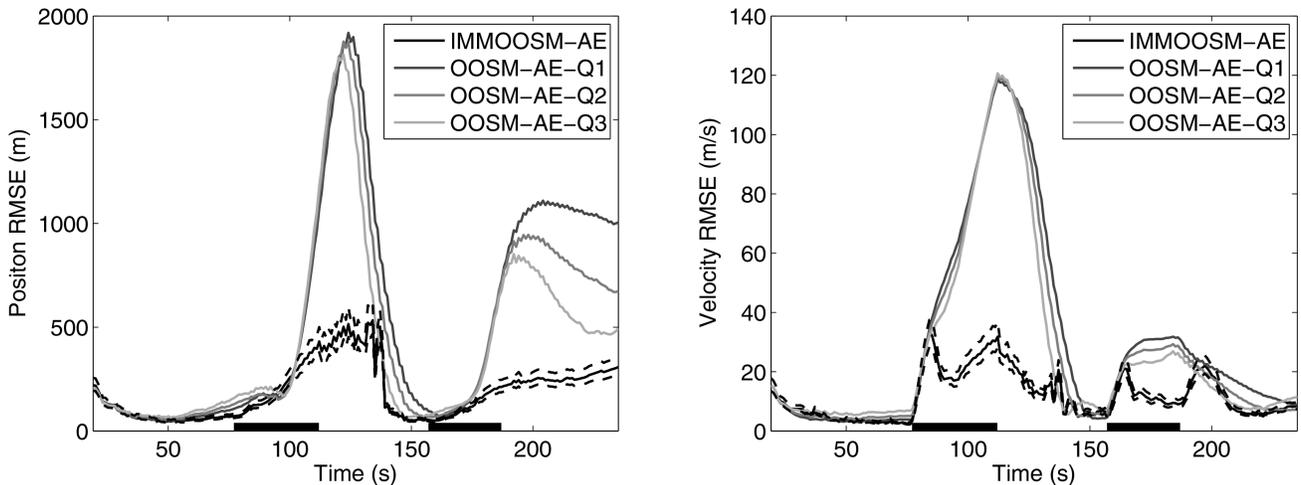


Fig. 8. Average position RMSE (left) and average velocity RMSE (right) for the S-turn scenario. The dashed lines around the IMMOOSM-AE curve represent the variability (2σ) of its performance.

single-model filter, OOSM-AE-Q3, is able to track the maneuvering target better than its lower process noise counterparts.

The position and velocity RMSE from 100 Monte Carlos runs for the U-turn scenario for OOSM-AE-Q1, OOSM-AE-Q2, OOSM-AE-Q3 and IMMOOSM-

AE are shown in Fig. 6. The dashed lines indicate the 95% confidence region for RMSE for IMMOOSM-AE. It can be seen that the variability of the performance of the IMMOOSM-AE is much smaller than the difference between it and the performance of the OOSM-AE filters. The maneuvering interval is shown as a thicker line on the time axis.

For a single run of the S-turn scenario, the estimated trajectories with OOSM-AE-Q1, OOSM-AE-Q2, OOSM-AE-Q3 and IMMOOSM-AE are shown in Fig 7. It can be observed in this particular run that none of the single-model filters are able to cope with the first sharp turn (4.5 deg/s). On the other hand, the IMM filter is able to track the maneuvering targets through both turns.

The position and velocity RMSE from 100 Monte Carlos runs for the S-turn scenario for OOSM-AE-Q1, OOSM-AE-Q2, OOSM-AE-Q3 and IMMOOSM-AE are shown in Fig. 8. The dashed lines indicate the 95% confidence region for RMSE for IMMOOSM-AE. The maneuvering intervals are shown as thicker lines on the time axis.

VI. CONCLUSIONS

The IMMOOSM-AE estimator is capable of tracking a maneuvering target by fusing the measurements from an EO (or ESM) sensor and the delayed measurements from an acoustic sensor when both are on the same stationary platform. As demonstrated in the test scenarios, the estimation accuracy in terms of RMSE is improved significantly over the single-model based OOSM-AE.

APPENDIX A GHM FOR CV

The GHM transition model, g^{CV} , for the CV model is

$$g^{CV}[\hat{\mathbf{x}}^A(t_j^{e2} | t_k^{s1}), \hat{\mathbf{x}}^E(t_k^{s1})] = [g_1^{CV}(\cdot) \quad g_2^{CV}(\cdot) \quad g_3^{CV}(\cdot) \quad g_4^{CV}(\cdot) \quad g_5^{CV}(\cdot) \quad g_6^{CV}(\cdot)]' \quad (55)$$

where

$$g_1^{CV} = x(t_j^{e2}) - x(t_k^{s1}) - \dot{x}(t_k^{s1})T_{j,k} \quad (56)$$

$$g_2^{CV} = y(t_j^{e2}) - y(t_k^{s1}) - \dot{y}(t_k^{s1})T_{j,k} \quad (57)$$

$$g_3^{CV} = \dot{x}(t_j^{e2}) - \dot{x}(t_k^{s1}) \quad (58)$$

$$g_4^{CV} = \dot{y}(t_j^{e2}) - \dot{y}(t_k^{s1}) \quad (59)$$

$$g_5^{CV} = 0 \quad (60)$$

$$g_6^{CV} = t_j^{e2} + \frac{r_{j,\ell}}{c^p} - t_k^{s2} \quad (61)$$

and

$$T_{j,k} = t_j^{e2} - t_k^{s1} < 0 \quad (62)$$

$$\begin{aligned} r_{j,\ell} &= \sqrt{[x(t_j^{e2}) - x^s(t_\ell^{s2})]^2 + [y(t_j^{e2}) - y^s(t_\ell^{s2})]^2} \\ &= \sqrt{x(t_j^{e2})^2 + y(t_j^{e2})^2} \end{aligned} \quad (63)$$

The Jacobian matrix, \mathbf{A}^{CV} , for the CV model is

$$\mathbf{A}^{CV} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \frac{\partial g_1^{CV}}{\partial t_j^{e2}} \\ 0 & 1 & 0 & 0 & 0 & \frac{\partial g_2^{CV}}{\partial t_j^{e2}} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{\partial g_6^{CV}}{\partial x(t_j^{e2})} & \frac{\partial g_6^{CV}}{\partial y(t_j^{e2})} & 0 & 0 & 0 & 1 \end{bmatrix} \quad (64)$$

where

$$\frac{\partial g_1^{CV}}{\partial t_j^{e2}} = -\dot{x}(t_k^{s1}) \quad (65)$$

$$\frac{\partial g_2^{CV}}{\partial t_j^{e2}} = -\dot{y}(t_k^{s1}) \quad (66)$$

$$\begin{aligned} \frac{\partial g_6^{CV}}{\partial x(t_j^{e2})} &= \frac{[x(t_j^{e2} | t_k^{s1})]^p - x^s(t_\ell^{s2})}{c^p \sqrt{[x(t_j^{e2} | t_k^{s1})]^p - x^s(t_\ell^{s2})]^2 + [y(t_j^{e2} | t_k^{s1})]^p - y^s(t_\ell^{s2})]^2}} \\ &= \frac{[x(t_j^{e2} | t_k^{s1})]^p}{c^p \sqrt{[x(t_j^{e2} | t_k^{s1})]^p]^2 + [y(t_j^{e2} | t_k^{s1})]^p]^2}} \end{aligned} \quad (67)$$

$$\begin{aligned} \frac{\partial g_6^{CV}}{\partial y(t_j^{e2})} &= \frac{[y(t_j^{e2} | t_k^{s1})]^p - y^s(t_\ell^{s2})}{c^p \sqrt{[x(t_j^{e2} | t_k^{s1})]^p - x^s(t_\ell^{s2})]^2 + [y(t_j^{e2} | t_k^{s1})]^p - y^s(t_\ell^{s2})]^2}} \\ &= \frac{[y(t_j^{e2} | t_k^{s1})]^p}{c^p \sqrt{[x(t_j^{e2} | t_k^{s1})]^p]^2 + [y(t_j^{e2} | t_k^{s1})]^p]^2}} \end{aligned} \quad (68)$$

APPENDIX B GHM FOR CT

The GHM transition model, g^{CT} , for the CT model is

$$g^{CT}[\hat{\mathbf{x}}^A(t_j^{e2} | t_k^{s1}), \hat{\mathbf{x}}^E(t_k^{s1})] = [g_1^{CT}(\cdot) \quad g_2^{CT}(\cdot) \quad g_3^{CT}(\cdot) \quad g_4^{CT}(\cdot) \quad g_5^{CT}(\cdot) \quad g_6^{CT}(\cdot)]' \quad (69)$$

where

$$\begin{aligned} g_1^{CT} &= x(t_j^{e2}) - x(t_k^{s1}) - \frac{\sin[\omega(t_k^{s1})T_{j,k}]}{\omega(t_k^{s1})} \dot{x}(t_k^{s1}) \\ &\quad + \frac{1 - \cos[\omega(t_k^{s1})T_{j,k}]}{\omega(t_k^{s1})} \dot{y}(t_k^{s1}) \end{aligned} \quad (70)$$

$$\begin{aligned} g_2^{CT} &= y(t_j^{e2}) - y(t_k^{s1}) - \frac{1 - \cos[\omega(t_k^{s1})T_{j,k}]}{\omega(t_k^{s1})} \dot{x}(t_k^{s1}) \\ &\quad - \frac{\sin[\omega(t_k^{s1})T_{j,k}]}{\omega(t_k^{s1})} \dot{y}(t_k^{s1}) \end{aligned} \quad (71)$$

$$g_3^{\text{CT}} = \dot{x}(t_j^{\ell_2}) - \cos[\omega(t_k^{s_1})T_{j,k}] \dot{x}(t_k^{s_1}) + \sin[\omega(t_k^{s_1})T_{j,k}] \dot{y}(t_k^{s_1}) \quad (72)$$

$$g_4^{\text{CT}} = \dot{y}(t_j^{\ell_2}) - \sin[\omega(t_k^{s_1})T_{j,k}] \dot{x}(t_k^{s_1}) - \cos[\omega(t_k^{s_1})T_{j,k}] \dot{y}(t_k^{s_1}) \quad (73)$$

$$g_5^{\text{CT}} = \omega(t_j^{\ell_2}) - \omega(t_k^{s_1}) \quad (74)$$

$$g_6^{\text{CT}} = t_j^{\ell_2} + \frac{r_{j,\ell}}{c^p} - t_k^{s_2} \quad (75)$$

and

$$T_{j,k} = t_j^{\ell_2} - t_k^{s_1} < 0 \quad (76)$$

$$r_{j,\ell} = \sqrt{x(t_j^{\ell_2})^2 + y(t_j^{\ell_2})^2} \quad (77)$$

The Jacobian matrix, \mathbf{A}^{CT} , for the CT model is

$$\mathbf{A}^{\text{CT}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \frac{\partial g_1^{\text{CT}}}{\partial t_j^{\ell_2}} \\ 0 & 1 & 0 & 0 & 0 & \frac{\partial g_2^{\text{CT}}}{\partial t_j^{\ell_2}} \\ 0 & 0 & 1 & 0 & 0 & \frac{\partial g_3^{\text{CT}}}{\partial t_j^{\ell_2}} \\ 0 & 0 & 0 & 1 & 0 & \frac{\partial g_4^{\text{CT}}}{\partial t_j^{\ell_2}} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{\partial g_6^{\text{CT}}}{\partial x(t_j^{\ell_2})} & \frac{\partial g_6^{\text{CT}}}{\partial y(t_j^{\ell_2})} & 0 & 0 & 0 & 1 \end{bmatrix} \quad (78)$$

where

$$\frac{\partial g_1^{\text{CT}}}{\partial t_j^{\ell_2}} = -\dot{x}(t_k^{s_1}) \cos[\omega(t_k^{s_1})T_{j,k}] + \dot{y}(t_k^{s_1}) \sin[\omega(t_k^{s_1})T_{j,k}] \quad (79)$$

$$\frac{\partial g_2^{\text{CT}}}{\partial t_j^{\ell_2}} = -\dot{x}(t_k^{s_1}) \sin[\omega(t_k^{s_1})T_{j,k}] - \dot{y}(t_k^{s_1}) \cos[\omega(t_k^{s_1})T_{j,k}] \quad (80)$$

$$\frac{\partial g_3^{\text{CT}}}{\partial t_j^{\ell_2}} = \omega(t_k^{s_1}) \dot{x}(t_k^{s_1}) \sin[\omega(t_k^{s_1})T_{j,k}] + \omega(t_k^{s_1}) \dot{y}(t_k^{s_1}) \cos[\omega(t_k^{s_1})T_{j,k}] \quad (81)$$

$$\frac{\partial g_4^{\text{CT}}}{\partial t_j^{\ell_2}} = -\omega(t_k^{s_1}) \dot{x}(t_k^{s_1}) \cos[\omega(t_k^{s_1})T_{j,k}] + \omega(t_k^{s_1}) \dot{y}(t_k^{s_1}) \sin[\omega(t_k^{s_1})T_{j,k}] \quad (82)$$

$$\begin{aligned} & \frac{\partial g_6^{\text{CT}}}{\partial x(t_j^{\ell_2})} \\ &= \frac{[[x(t_j^{\ell_2} | t_k^{s_1})]^p - x^s(t_k^{s_2})]}{c^p \sqrt{[[x(t_j^{\ell_2} | t_k^{s_1})]^p - x^s(t_k^{s_2})]^2 + [[y(t_j^{\ell_2} | t_k^{s_1})]^p - y^s(t_k^{s_2})]^2}} \\ &= \frac{[x(t_j^{\ell_2} | t_k^{s_1})]^p}{c^p \sqrt{[[x(t_j^{\ell_2} | t_k^{s_1})]^p]^2 + [[y(t_j^{\ell_2} | t_k^{s_1})]^p]^2}} \end{aligned} \quad (83)$$

$$\begin{aligned} & \frac{\partial g_6^{\text{CT}}}{\partial y(t_j^{\ell_2})} \\ &= \frac{[[y(t_j^{\ell_2} | t_k^{s_1})]^p - y^s(t_k^{s_2})]}{c^p \sqrt{[[x(t_j^{\ell_2} | t_k^{s_1})]^p - x^s(t_k^{s_2})]^2 + [[y(t_j^{\ell_2} | t_k^{s_1})]^p - y^s(t_k^{s_2})]^2}} \\ &= \frac{[y(t_j^{\ell_2} | t_k^{s_1})]^p}{c^p \sqrt{[[x(t_j^{\ell_2} | t_k^{s_1})]^p]^2 + [[y(t_j^{\ell_2} | t_k^{s_1})]^p]^2}} \end{aligned} \quad (84)$$

and

$$[T_{j,k}]^p = [t_j^{\ell_2}]^p - t_k^{s_1} \quad (85)$$

APPENDIX C CONFIDENCE REGION FOR TRUE RMSE

The position or velocity error, given by $\hat{\mathbf{x}} - \mathbf{x}^g$, is assumed to follow a zero-mean Gaussian distribution with unknown variance, RMSE^2 . N independent Monte Carlo observations are taken of this error, $\hat{\mathbf{x}}(n) - \mathbf{x}^g$, $n = 1, \dots, N$. The maximum likelihood estimator, $\overline{\text{RMSE}}^2(N)$, of RMSE^2 is thus given by

$$\overline{\text{RMSE}}^2(N) = \frac{1}{N} \sum_{n=1}^N (\hat{\mathbf{x}}(n) - \mathbf{x}^g)^2 \quad (86)$$

The square $\overline{\text{RMSE}}^2(N)$ follows a scaled chi-squared distribution with N degrees of freedom, i.e. $\overline{\text{RMSE}}^2(N) \sim (\text{RMSE}^2/N) \chi_N^2$. Note that the position and velocity errors are 2-dimensional. However, the x and y errors are correlated, so the number of degrees of freedom is somewhere between N and $2N$. To be conservative, N is chosen, which will give a larger confidence region in the sequel.

The 95% probability interval for the ratio $\overline{\text{RMSE}}^2(N)/\text{RMSE}^2$ for $N = 100$ is given below.

$$\begin{aligned} \frac{\overline{\text{RMSE}}^2(N)}{\text{RMSE}^2} &\in \left[\left(\frac{1}{100} \right) \chi_N^2(2.5\%), \left(\frac{1}{100} \right) \chi_N^2(97.5\%) \right] \\ &= [0.74, 1.3] \end{aligned} \quad (87)$$

The 95% confidence region for RMSE^2 given $\overline{\text{RMSE}}^2(N)$ is thus

$$\frac{\text{RMSE}^2}{\overline{\text{RMSE}}^2(N)} \in [(0.74)^{-1}, (1.3)^{-1}] = [0.77, 1.35] \quad (88)$$

Therefore, the 95% confidence region for RMSE given $\overline{\text{RMSE}}(N)$ is

$$\frac{\text{RMSE}}{\overline{\text{RMSE}}(N)} \in [\sqrt{0.77}, \sqrt{1.35}] = [0.88, 1.16] \quad (89)$$

i.e.,

$$\text{RMSE} \in [0.88 \overline{\text{RMSE}}(N), 1.16 \overline{\text{RMSE}}(N)] \quad (90)$$

REFERENCES

- [1] V. Aidala
 “Kalman filter behavior in bearings-only tracking applications,”
Aerospace and Electronic Systems, IEEE Transactions on, vol. AES-15, no. 1, pp. 29–39, Jan 1979.
- [2] Y. Bar-Shalom
 “Update with out-of-sequence measurements in tracking: exact solution,”
Aerospace and Electronic Systems, IEEE Transactions on, vol. 38, no. 3, pp. 769–777, Jul 2002.
- [3] Y. Bar-Shalom, M. Mallick, H. Chen, and R. Washburn
 “One-step solution for the general out-of-sequence-measurement problem in tracking,”
 in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 4, 2002, pp. 4-1551–4-1559 vol.4.
- [4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.
 John Wiley & Sons, 2001.
- [5] Y. Bar-Shalom, P. K. Willett, and X. Tian
Tracking and Data Fusion: A Handbook of Algorithms.
 YBS Publishing, 2011.
- [6] H. Blom and Y. Bar-Shalom
 “The interacting multiple model algorithm for systems with markovian switching coefficients,”
Automatic Control, IEEE Transactions on, vol. 33, no. 8, pp. 780–783, Aug 1988.
- [7] J. E. L. Cadre and C. Jauffret
 “Discrete-time observability and estimability analysis for bearings-only target motion analysis,”
IEEE Transactions on Aerospace and Electronic Systems, vol. 33, no. 1, pp. 178–201, Jan 1997.
- [8] E. Fogel and M. Gavish
 “Nth-order dynamics target observability from angle measurements,”
Aerospace and Electronic Systems, IEEE Transactions on, vol. 24, no. 3, pp. 305–308, May 1988.
- [9] C. Jauffret, D. Pillon, and A.-C. Perez-Pignol
 “Bearings-only TMA without observer maneuver,”
 in *Information Fusion, Proc. 11th International Conference on*, June 2008.
- [10] S. J. Julier and J. K. Uhlmann
 “New extension of the Kalman filter to nonlinear systems,”
 in *Proc. SPIE, Conf. Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068, Orlando, FL, April 1997.
- [11] A. G. Lindgren and K. F. Gong
 “Position and velocity estimation via bearing observations,”
Aerospace and Electronic Systems, IEEE Transactions on, no. 4, pp. 564–577, 1978.
- [12] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan
 “Interacting multiple model methods in target tracking: a survey,”
Aerospace and Electronic Systems, IEEE Transactions on, vol. 34, no. 1, pp. 103–123, Jan. 1998. [Online]. Available: <http://dx.doi.org/10.1109/7.640267>.
- [13] U. Orguner and F. Gustafsson
 “Target tracking using delayed measurements with implicit constraints,”
 in *Information Fusion, Proc. 11th International Conference on*, June 2008.
- [14] ———
 “Particle filtering with propagation delayed measurements,”
 in *Proc. Aerospace Conference, 2010 IEEE*, Big Sky, MT, March 2010.
- [15] A. Papoulis and S. Pillai
Probability, Random Variables, and Stochastic Processes, ser. McGraw-Hill series in electrical and computer engineering. McGraw-Hill, 2002. [Online]. Available: <https://books.google.com.sg/books?id=WM69jwEACAAJ>.
- [16] A. Petersen and R. Koch
 “Statistical analysis of Kalman filters by conversion to Gauss-Helmert models with applications to process noise estimation,”
 in *Proc. Pattern Recognition (ICPR), 2010 20th International Conference on*, Aug 2010, pp. 2386–2389.
- [17] R. Yang, Y. Bar-Shalom, and G. W. Ng
 “Bearings-only tracking with fusion from heterogenous passive sensors: ESM/EO and acoustic,”
 in *Proc. 18th Intn’l Conf. on Information Fusion*, Washington, DC, July 2015.
- [18] R. Yang, Y. Bar-Shalom, H. A. J. Huang, and G. W. Ng
 “UGHF for acoustic tracking with state-dependent propagation delay,”
 vol. 51, no. 3, July 2015, pp. 1747–1761.
- [19] R. Yang, J. Huang, G. W. Ng, and Y. Bar-Shalom
 “Interacting multiple model unscented Gauss-Helmert filter for bearings-only tracking with state-dependent propagation delay,”
 in *Information Fusion (FUSION), Proc. 17th International Conference on*, July 2014.
- [20] T. Yuan, Y. Bar-Shalom, P. Willett, E. Mozeson, S. Pollak, and D. Hardiman
 “A multiple IMM estimation approach with unbiased mixing for thrusting projectiles,”
Aerospace and Electronic Systems, IEEE Transactions on, vol. 48, no. 4, pp. 3250–3267, October 2012.



Hong An Jack Huang was born in Singapore in 1983. He received a B.E. degree from National University of Singapore (NUS) in 2008. He is currently a senior member of technical staff at DSO National Laboratories, Singapore. His research interests in target tracking include GMTI tracking, passive tracking, and image tracking.

Yaakov Bar-Shalom was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in electrical engineering. From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, California. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. He is also Director of the ESP (Estimation and Signal Processing) Lab. His current research interests are in estimation theory, target tracking and data fusion. He has published over 500 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate texts *Estimation and Tracking: Principles, Techniques and Software* (Artech House, 1993; translated into Russian, MGTU Bauman, Moscow, Russia, 2011), *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction* (Wiley, 2001), the advanced graduate texts *Multitarget-Multisensor Tracking: Principles and Techniques* (YBS Publishing, 1995), *Tracking and Data Fusion* (YBS Publishing, 2011), and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Artech House, Vol. I, 1990; Vol. II, 1992; Vol. III, 2000). He has been elected Fellow of IEEE for “contributions to the theory of stochastic systems and of multi-target tracking.” He has been consulting to numerous companies and government agencies, and originated the series of Multitarget-Multisensor Tracking short courses offered via UCLA Extension, at Government Laboratories, private companies and overseas. During 1976 and 1977 he served as Associate Editor of the IEEE Transactions on Automatic Control and from 1978 to 1981 as Associate Editor of Automatica. He was Program Chairman of the 1982 American Control Conference, General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–87 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–89 was a member of the Board of Governors of the IEEE CSS. He was a member of the Board of Directors of the International Society of Information Fusion (1999–2004) and served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications in 2004–13. In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS and has given numerous keynote addresses at major national and international conferences. He is co-recipient of the M. Barry Carlton Award for the best paper in the IEEE Transactions on Aerospace and Electronic Systems in 1995 and 2000 and recipient of the 1998 University of Connecticut AAUP Excellence Award for Research. 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. In 2015, he received from the International Society for Information Fusion the Lifetime of Excellence in Information Fusion award. Since 2016, this award has been renamed as the Yaakov Bar-Shalom Award for Lifetime of Excellence in Information Fusion. 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.





Rong Yang received her B.E. degree in information and control from Xi'an Jiao Tong University, China in 1986, M.Sc. degree in electrical engineering from the National University of Singapore in 2000, and Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore in 2012. She is currently a Principal Member of Technical Staff at DSO National Laboratories, Singapore. Her research interests include passive tracking, low observable target tracking, GMTI tracking, hybrid dynamic estimation and data fusion. She received the FUSION 2014 Best Paper Award (First runner up).



Gee Wah Ng received his M.Sc. and Ph.D. from University of Manchester Institute of Science and Technology, United Kingdom. He is currently a Distinguished Member of Technical Staff and Programme Director (Information Exploitation) of Information Division at DSO National Laboratories. He has delivered many projects in the decision support areas and has authored three books. He is active in international conferences in the areas of information fusion and intelligent systems. His research interests in data and information fusion include target tracking, computational intelligence, machine learning, self-tuning and sensor networks.

JAIIF

Journal of Advances in Information Fusion

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

Volumes 1–11 Index

- Adurthi, N.**, *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.
- Ala-Luhtala, J.**, *see* Raitoharju, M., *JAIF*, **11**, 1 (June 2016), 3–14.
- Alford, M.**, *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.
- Ali-Löytty, S.**, *see* Raitoharju, M., *JAIF*, **11**, 1 (June 2016), 3–14.
- Alsun, M.**, Lecornu, L., Solaiman, B., Possibilistic Medical Knowledge Representation Model, *JAIF*, **7**, 2 (December 2012), 101–113.
- An, W.**, Singh, S., Pattipati, K. R., Kleinman, D. L., and Gokhale, S. S., Dynamic Scheduling of Multiple Hidden Markov Model-Based Sensors, *JAIF*, **3**, 1 (July 2008), 33–49.
- Anderson, S. L.**, *see* Stone, L. D., *JAIF*, **10**, 1 (June 2015), 3–12.
- Anderl, S. F.**, *see* Karlsson, A., *JAIF*, **6**, 2 (December 2011), 150–166.
- Aravinthan, A.**, *see* Habtemariam, B. K., *JAIF*, **7**, 2 (December 2012), 114–130.
- Areta, J.**, Bar-Shalom, Y., and Pattipati, K. R., T2T and M2T Association with Combined Hypotheses, *JAIF*, **4**, 1 (July 2009), 40–51.
- Areta, J.**, Bar-Shalom, Y., and Rothrock, R., Misassociation Probability in M2TA and T2TA, *JAIF*, **2**, 2 (Dec. 2007), 113–127.
- Areta, J.**, Bar-Shalom, Y., Levedahl, M., and Pattipati, K. R., Hierarchical Track Association and Fusion for a Networked Surveillance System, *JAIF*, **1**, 2 (Dec. 2006), 144–157.
- Arnborg, S.**, *see* Brynielsson, J., *JAIF*, **1**, 2 (Dec. 2006), 108–121.
- Arnborg, S.**, Robust Bayesianism: Relation to Evidence Theory *JAIF*, **1**, 1 (July 2006), 75–90.
- Aughenbaugh, J. M.** and La Cour, B. R., Measurement-Guided Likelihood Sampling for Grid-Based Bayesian Tracking *JAIF*, **5**, 2 (Dec. 2010), 108–127.
- Avasara, V.**, Mullen, T., and Hall, D., A Market-based Approach to Sensor Management, *JAIF*, **4**, 1 (July 2009), 52–71.

B

- Bab-Hadiashar, A.**, *see* Hoseinnezhad, R., *JAIF*, **1**, 1 (July 2006), 52–62.
- Balasingam, B.**, *see* Choi, S., *JAIF*, **8**, 2 (December 2013), 143–155.
- Bar-Shalom, Y.**, *see* Areta, J., *JAIF*, **1**, 2 (Dec. 2006), 144–157.
- Bar-Shalom, Y.**, *see* Areta, J., *JAIF*, **2**, 2 (Dec. 2007), 113–127.
- Bar-Shalom, Y.**, *see* Areta, J., *JAIF*, **4**, 1 (July 2009), 40–51.
- Bar-Shalom, Y.**, *see* Belfadel, D., *JAIF*, **9**, 2 (December 2014), 59–74.
- Bar-Shalom, Y.**, *see* Belfadel, D., *JAIF*, **10**, 2 (December 2015), 101–112.
- Bar-Shalom, Y.** and Chen, H., Covariance Reconstruction for Track Fusion with Legacy Track Sources *JAIF*, **3**, 2 (Dec. 2008), 107–117.
- Bar-Shalom, Y.** and Chen, H., Multisensor Track-to-Track Association for Tracks with Dependent Errors *JAIF*, **1**, 1 (July 2006), 3–14.
- Bar-Shalom, Y.** and Chen, H., Track-to-Track Association Using Attributes *JAIF*, **2**, 1 (July 2007), 49–59.
- Bar-Shalom, Y.**, *see* Crouse, D. F., *JAIF*, **8**, 1 (July 2013), 73–89.
- Bar-Shalom, Y.**, *see* Dou, W., *JAIF*, **10**, 2 (December 2015), 163–182.
- Bar-Shalom, Y.**, *see* Osborne, R. W., III, *JAIF*, **7**, 1 (June 2012), 3–15.
- Bar-Shalom, Y.**, *see* Osborne, R. W., III, *JAIF*, **9**, 2 (December 2014), 75–89.
- Bar-Shalom, Y.**, *see* Osborne, III, R. W., *JAIF*, **10**, 2 (December 2015), 199–210.
- Bar-Shalom, Y.**, *see* Ravindra, V. C., *JAIF*, **5**, 2 (Dec. 2010), 88–107.
- Bar-Shalom, Y.**, *see* Rodningsby, A., *JAIF*, **4**, 2 (Dec. 2009), 117–145.
- Bar-Shalom, Y.**, *see* Romeo, K., *JAIF*, **10**, 2 (December 2015), 113–124.
- Bar-Shalom, Y.**, *see* Tharmarasa, R., *JAIF*, **7**, 1 (June 2012), 46–60.
- Bar-Shalom, Y.**, *see* Tian, X., *JAIF*, **4**, 2 (Dec. 2009), 146–164.
- Bar-Shalom, Y.**, *see* Tian, X., *JAIF*, **5**, 1 (July 2010), 3–17.
- Bar-Shalom, Y.**, *see* Tian, X., *JAIF*, **5**, 2 (Dec. 2010), 128–138.
- Bar-Shalom, Y.**, *see* Yuan, T., *JAIF*, **6**, 2 (December 2011), 131–149.
- Bar-Shalom, Y.**, *see* Zhang, S., *JAIF*, **6**, 1 (June 2011), 3–23.
- Bar-Shalom, Y.**, *see* Zhang, S., *JAIF*, **9**, 1 (July 2014), 38–46.
- Baum, M.**, *see* Faion, F., *JAIF*, **10**, 1 (June 2015), 13–30.
- Belaroussi, R.**, Prevost, L., and Milgram, M., Algorithms Fusion for Face Localization, *JAIF*, **1**, 1 (July 2006), 35–51.
- Belfadel, D.**, Osborne, R. W., III, Bar-Shalom, Y., Bias Estimation and Observability for Optical Sensor Measurements with Targets of Opportunity, *JAIF*, **9**, 2 (December 2014), 59–74.
- Belfadel, D.**, Osborne, R. W., III, Bar-Shalom, Y., Bias Estimation for Moving Optical Sensor Measurements with Targets of Opportunity, *JAIF*, **10**, 2 (December 2015), 101–112.
- Benaskeur, A. R.**, Rhéaume, F., and Paradis, S., Target Engageability Improvement through Adaptive Tracking, *JAIF*, **2**, 2 (Dec. 2007), 99–112.
- Biermann, J.**, Hörling, P., Snidaro, L., Experiences and Challenges in Automated Support for Intelligence in Asymmetric Operations, *JAIF*, **8**, 2 (December 2013), 101–118.
- Blair, W. D.** and Miceli, P. A., Performance Prediction of Multisensor Tracking Systems for Single Maneuvering Targets *JAIF*, **7**, 1 (June 2012), 28–45.
- Blasch, E.**, *see* Chen, G., *JAIF*, **2**, 1 (July 2007), 35–48.
- Blasch, E.**, *see* Duník, J., *JAIF*, **11**, 1 (June 2016), 91–109.

- Blasch, E.**, Kadar, I., Salerno, J., Kokar, M. M., Das, S., Powell, G. M., Corkill, D. D., and Ruspini, E. H., Issues and Challenges in Situation Assessment (Level 2 Fusion), *JAIF*, **1**, 2 (Dec. 2006), 122–143.
- Blasch, E.**, *see* Kahler, B., *JAIF*, **6**, 2 (December 2011), 101–118.
- Blasch, E.**, *see* Yang, C., *JAIF*, **3**, 1 (July 2008), 14–32.
- Blasch, E.**, *see* Zheng, Y., *JAIF*, **9**, 2 (December 2014), 124–135.
- Bluem, E. A.**, *see* Blom, H. A. P., *JAIF*, **1**, 1 (July 2006), 15–34.
- Blom, H. A. P.** and Bluem, E. A., Joint Particle Filtering of Multiple Maneuvering Targets From Unassociated Measurements *JAIF*, **1**, 1 (July 2006), 15–34.
- Bossé, É.**, *see* Valin, P., *JAIF*, **5**, 1 (July 2010), 32–40.
- Brynielsson, J.** and Arnborg, S., An Information Fusion Game Component *JAIF*, **1**, 2 (Dec. 2006), 108–121.
- Bubalo, A.**, *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.
- Bukal, M.**, *see* Marković, I., *JAIF*, **11**, 2 (December 2016), 157–172.
- Bursik, M. I.**, *see* Rogova, G. L., *JAIF*, **3**, 2 (Dec. 2008), 118–128.

C

- Carthel, C.**, *see* Coraluppi, S., *JAIF*, **5**, 1 (July 2010), 18–31.
- Carthel, C.**, *see* Coraluppi, S., *JAIF*, **6**, 1 (June 2011), 57–67.
- Carthel, C.**, *see* Coraluppi, S., *JAIF*, **7**, 2 (December 2012), 153–164.
- Ćesić, J.**, *see* Marković, I., *JAIF*, **11**, 2 (December 2016), 157–172.
- Chakravorty, R.** and Challa, S., Augmented State Integrated Probabilistic Data Association Smoothing for Automatic Track Initiation in Clutter *JAIF*, **1**, 1 (July 2006), 63–74.
- Carvalho, R. N.** and Chang, K., A Fusion Analysis and Evaluation Tool for Multi-Sensor Classification Systems *JAIF*, **7**, 2 (December 2012), 141–152.
- Challa, S.**, *see* Chakravorty, R., *JAIF*, **1**, 1 (July 2006), 63–74.
- Chang, K.**, *see* Carvalho, R. N., *JAIF*, **7**, 2 (December 2012), 141–152.
- Chang, K. C.** and Hill, J. P., Level I and Level II Target Valuations for Sensor Management *JAIF*, **1**, 2 (Dec. 2006), 95–107.
- Chen, G.**, Shen, D., Kwan, C., Cruz, J. B., Jr., Kruger, M., and Blasch, E., Game Theoretic Approach to Threat Prediction and Situation Awareness, *JAIF*, **2**, 1 (July 2007), 35–48.
- Chen, H.**, *see* Bar-Shalom, Y., *JAIF*, **1**, 1 (July 2006), 3–14.
- Chen, H.**, *see* Bar-Shalom, Y., *JAIF*, **2**, 1 (July 2007), 49–59.
- Chen, H.**, *see* Bar-Shalom, Y., *JAIF*, **3**, 2 (Dec. 2008), 107–117.
- Choi, S.**, Balasingam, B., Willett, P., Testing Under Communication Constraints, *JAIF*, **8**, 2 (December 2013), 143–155.
- Choi, S.**, Willett, P., Zhou, S., The PMHT for Passive Radar in a DAB/DVB Network, *JAIF*, **9**, 1 (July 2014), 27–37.
- Coraluppi, S.** and Carthel, C., Multi-Stage Multiple-Hypothesis Tracking *JAIF*, **6**, 1 (June 2011), 57–67.
- Coraluppi, S.** and Carthel, C., Modified Scoring in Multiple-Hypothesis Tracking *JAIF*, **7**, 2 (December 2012), 153–164.
- Coraluppi, S.**, *see* Erdinc, O., *JAIF*, **2**, 1 (July 2007), 22–34.
- Coraluppi, S.**, Guerriero, M., Willett, P., and Carthel, C., Fuse-before-Track in Large Sensor Networks, *JAIF*, **5**, 1 (July 2010), 18–31.
- Corkill, D. D.**, *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.
- Cornacchia, M.**, *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.
- Crassidis, J. L.**, *see* George, J., *JAIF*, **6**, 1 (June 2011), 39–56.
- Crouse, D. F.**, Guerriero, M., and Willett, P., A Critical Look at the PMHT, *JAIF*, **4**, 2 (Dec. 2009), 93–116.
- Crouse, D. F.**, Osborne, R. W., III, Pattipati, K., Willett, P., Bar-Shalom, Y., Efficient 2D Sensor Location Estimation using Targets of Opportunity, *JAIF*, **8**, 1 (July 2013), 73–89.
- Crouse, D. F.**, Simulating Aerial Targets in 3D Accounting for the Earth's Curvature *JAIF*, **10**, 1 (June 2015), 31–57.
- Crouse, D. F.**, Zheng, L., Willett, P., Corrections to "A Critical Look at the PMHT", *JAIF*, **7**, 1 (June 2012), 97–98.
- Cruz, J. B., Jr.**, *see* Chen, G., *JAIF*, **2**, 1 (July 2007), 35–48.

D

- Damarla, T.**, *see* Ravindra, V. C., *JAIF*, **5**, 2 (Dec. 2010), 88–107.
- Darling, J. E.** and Demars, K. Ju., Uncertainty Propagation of Correlated Quaternion and Euclidean States Using the Gauss-Bingham Density *JAIF*, **11**, 2 (December 2016), 186–205.
- Das, S.**, *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.
- Davey, S. J.**, Detecting a Small Boat using Histogram PMHT *JAIF*, **6**, 2 (December 2011), 167–186.
- Degen, C.**, Govaers, F., Koch, W., Tracking Targets with Multiple Measurements per Scan Using the Generalized PHD Filter, *JAIF*, **10**, 2 (December 2015), 125–142.
- Demars, K. Ju.**, *see* Darling, J. E., *JAIF*, **11**, 2 (December 2016), 186–205.
- Dezert, J.**, *see* Martin, A., *JAIF*, **3**, 2 (Dec. 2008), 67–89.
- Djiknavorian, P.**, *see* Valin, P., *JAIF*, **5**, 1 (July 2010), 32–40.

- Dou, W.**, Bar-Shalom, Y., Willett, P., Bistatic Measurement Fusion from Multistatic Configurations for Air Collision Warning, *JAIF*, **10**, 2 (December 2015), 163–182.
- Duník, J.**, Straka, O., Šimandl, M., Blasch, E., Sigma-Point Set Rotation for Derivative-Free Filters in Target Tracking Applications, *JAIF*, **11**, 1 (June 2016), 91–109.

E

- Eilbert, J.**, *see* Schrag, R. C., *JAIF*, **2**, 2 (Dec. 2007), 77–98.
- Erdinc, O.**, Willett, P., and Coraluppi, S., Multistatic Sensor Placement: A Tracking Approach, *JAIF*, **2**, 1 (July 2007), 22–34.

F

- Faion, F.**, Zea, A., Baum, M., Hanebeck, U. D., Symmetries in Bayesian Extended Object Tracking, *JAIF*, **10**, 1 (June 2015), 13–30.
- Falkman, G.**, *see* Johansson, F., *JAIF*, **6**, 2 (December 2011), 187–199.
- Ferry, J. P.**, Exact Association Probability for Data with Bias and Features *JAIF*, **5**, 1 (July 2010), 41–67.
- Flamant, J.**, *see* Le Bihan, N., *JAIF*, **11**, 2 (December 2016), 173–185.
- Foo, P. H.** and Ng, G. W., High-level Information Fusion: An Overview *JAIF*, **8**, 1 (July 2013), 33–72.
- Foo, P. H.**, Ng, G. W., Ng, K. H., and Yang, R., Application of Intent Inference for Air Defense and Conformance Monitoring, *JAIF*, **4**, 1 (July 2009), 3–26.
- Fosbury, A. M.**, *see* George, J., *JAIF*, **6**, 1 (June 2011), 39–56.

G

- George, J.**, Crassidis, J. L., Singh, T., Fosbury, A. M., Anomaly Detection using Context-Aided Target Tracking, *JAIF*, **6**, 1 (June 2011), 39–56.
- George, J.** and Kaplan, L. M., Shooter Localization using a Wireless Sensor Network of Soldier-Worn Gunfire Detection Systems *JAIF*, **8**, 1 (July 2013), 15–32.
- George, J.**, *see* Osborne, R. W., III, *JAIF*, **9**, 2 (December 2014), 75–89.
- Georgescu, R.**, Willett, P., Marano, S., Matta, V., Predetection Fusion in Large Sensor Networks with Unknown Target Locations, *JAIF*, **7**, 1 (June 2012), 61–77.
- Gilitschenski, I.**, *see* Kurz, G., *JAIF*, **9**, 2 (December 2014), 90–105.
- Gilitschenski, I.**, *see* Kurz, G., *JAIF*, **11**, 2 (December 2016), 138–156.
- Glattetre, J.**, *see* Rodningsby, A., *JAIF*, **4**, 2 (Dec. 2009), 117–145.
- Goger, P.**, *see* Schrag, R. C., *JAIF*, **2**, 2 (Dec. 2007), 77–98.
- Goker, A.**, *see* Kaliciak, L., *JAIF*, **10**, 2 (December 2015), 183–198.
- Gokhale, S. S.**, *see* An, W., *JAIF*, **3**, 1 (July 2008), 33–49.
- Govaers, F.**, *see* Degen, C., *JAIF*, **10**, 2 (December 2015), 125–142.
- Guerriero, M.**, *see* Coraluppi, S., *JAIF*, **5**, 1 (July 2010), 18–31.
- Gregory Watson, S.**, Zhang, S., *JAIF*, **6**, 1 (June 2011), 3–23.
- Guerriero, M.**, *see* Crouse, D. F., *JAIF*, **4**, 2 (Dec. 2009), 93–116.
- Guitouni, A.**, *see* Jabeur, K., *JAIF*, **4**, 2 (Dec. 2009), 75–92.
- Gustafsson, F.**, *see* Roth, M., *JAIF*, **11**, 1 (June 2016), 47–70.

H

- Habtemariam, B. K.**, Aravinthan, A., Tharmarasa, R., Punithakumar, K., Lang, T., Kirubarajan, T., Distributed Tracking with a PHD Filter using Efficient Measurement Encoding, *JAIF*, **7**, 2 (December 2012), 114–130.
- Hall, D.**, *see* Avasarala, V., *JAIF*, **4**, 1 (July 2009), 52–71.
- Hallingstad, O.**, *see* Rodningsby, A., *JAIF*, **4**, 2 (Dec. 2009), 117–145.
- Hamp, Q.** and Reindl, L., Association Performance Enhancement Through Classification *JAIF*, **7**, 2 (December 2012), 131–140.
- Han, X.**, Yu, F., Levchuck, G., Patipati, K., Tu, F., A Probabilistic Computational Model for Identifying Organizational Structures from Uncertain Activity Data, *JAIF*, **7**, 1 (June 2012), 78–96.
- Hanebeck, U. D.**, *see* Faion, F., *JAIF*, **10**, 1 (June 2015), 13–30.
- Hanebeck, U. D.**, *see* Kurz, G., *JAIF*, **9**, 1 (July 2014), 13–26.
- Hanebeck, U. D.**, *see* Kurz, G., *JAIF*, **9**, 2 (December 2014), 90–105.
- Hanebeck, U. D.**, *see* Kurz, G., *JAIF*, **11**, 2 (December 2016), 138–156.
- Hanebeck, U. D.**, *see* Pfaff, F., *JAIF*, **11**, 2 (December 2016), 206–226.
- Hanebeck, U. D.**, *see* Steinbring, J., *JAIF*, **9**, 2 (December 2014), 106–123.
- Hanebeck, U. D.**, *see* Steinbring, J., *JAIF*, **11**, 1 (June 2016), 71–90.
- Hanson-Hedgecock, S.**, *see* Rogova, G. L., *JAIF*, **3**, 2 (Dec. 2008), 118–128.
- Hartikainen, J.**, *see* Särkkä, S., *JAIF*, **11**, 1 (June 2016), 31–46.
- Hendeby, G.**, *see* Roth, M., *JAIF*, **11**, 1 (June 2016), 47–70.
- Hill, J. P.**, *see* Chang, K. C., *JAIF*, **1**, 2 (Dec. 2006), 95–107.
- Holender, M.**, *see* Papageorgiou, D. J., *JAIF*, **6**, 2 (December 2011), 77–100.
- Holender, M.**, *see* Sudit, M., *JAIF*, **2**, 1 (July 2007), 3–21.
- Hörling, P.**, *see* Biermann, J., *JAIF*, **8**, 2 (December 2013), 101–118.

- Hoseinnezhad, R.** and Bab-Hadiashar, A., Fusion of Redundant Information in Brake-By-Wire Systems Using a Fuzzy Voter *JAIF*, **1**, 1 (July 2006), 52–62.
- Howell, C.** and Moyer, S., Establishment of Human Performance Baseline for Image Fusion Algorithms in the LWIR and SWIR Spectra *JAIF*, **8**, 2 (December 2013), 133–142.

J

- Jabeur, K.** and Guitouni, A., A Generalized Framework for Multi-Criteria Classifiers with Automated Learning: Application on FLIR Ship Imagery *JAIF*, **4**, 2 (Dec. 2009), 75–92.
- Jauffret, C.**, Pillon, D., Pignol, A.-C., Leg-by-leg Bearings-Only Target Motion Analysis Without Observer Maneuver, *JAIF*, **6**, 1 (June 2011), 24–38.
- Jentschel, H.-J.**, *see* Junghans, M., *JAIF*, **3**, 1 (July 2008), 50–62.
- Jilkov, V. P.** and Li, X. R., On Fusion of Multiple Objectives for UAV Search & Track Path Optimization *JAIF*, **4**, 1 (July 2009), 27–39.
- Jilkov, V. P.** and Wu, J., Efficient GPU-Accelerated Implementation of Particle and Particle Flow Filters for Target Tracking *JAIF*, **10**, 1 (June 2015), 73–88.
- Johansson, F.** and Falkman, G., Real-time Allocation of Firing Units To Hostile Targets *JAIF*, **6**, 2 (December 2011), 187–199.
- Johansson, R.**, *see* Karlsson, A., *JAIF*, **6**, 2 (December 2011), 150–166.
- Julier, S.**, *see* Kurz, G., *JAIF*, **9**, 2 (December 2014), 90–105.
- Jones, E.**, *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.
- Junghans, M.** and Jentschel, H.-J., Correction of Selection Bias in Traffic Data by Bayesian Network Data Fusion *JAIF*, **3**, 1 (July 2008), 50–62.

K

- Kadar, I.**, *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.
- Kahler, B.** and Blasch, E., Decision-Level Fusion Performance Improvement From Enhanced HRR Radar Clutter Suppression *JAIF*, **6**, 2 (December 2011), 101–118.
- Kaliciak, L.**, Myrhaug, H., Goker, A., Song, D., Early Fusion and Query Modification in Their Dual Late Fusion Forms, *JAIF*, **10**, 2 (December 2015), 183–198.
- Kaplan, L. M.**, *see* George, J., *JAIF*, **8**, 1 (July 2013), 15–32.
- Kaplan, L.**, *see* Osborne, R. W., III, *JAIF*, **9**, 2 (December 2014), 75–89.
- Karlsson, A.**, Johansson, R., Andler, S. F., Characterization and Empirical Evaluation of Bayesian and Credal Combination Operators, *JAIF*, **6**, 2 (December 2011), 150–166.
- Kirubarajan, T.**, *see* Habtemariam, B. K., *JAIF*, **7**, 2 (December 2012), 114–130.
- Kirubarajan, T.**, *see* Tharmarasa, R., *JAIF*, **7**, 1 (June 2012), 46–60.
- Kleinman, D. L.**, *see* An, W., *JAIF*, **3**, 1 (July 2008), 33–49.
- Koch, W.**, *see* Degen, C., *JAIF*, **10**, 2 (December 2015), 125–142.
- Kokar, M. M.**, *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.
- Kokkala, J.**, Solin, A., Särkkä, S., Sigma-Point Filtering and Smoothing Based Parameter Estimation in Nonlinear Dynamic Systems, *JAIF*, **11**, 1 (June 2016), 15–30.
- Kroschel, K.**, *see* Schlosser, M. S., *JAIF*, **2**, 2 (Dec. 2007), 65–76.
- Kruger, M.**, *see* Chen, G., *JAIF*, **2**, 1 (July 2007), 35–48.
- Kurz, G.**, Gilitschenski, I., Julier, S., Hanebeck, U. D., Recursive Bingham Filter for Directional Estimation Involving 180 Degree Symmetry, *JAIF*, **9**, 2 (December 2014), 90–105.
- Kurz, G.**, Gilitschenski, I., Siegart, R. Y., Hanebeck, U. D., Methods for Deterministic Approximation of Circular Densities, *JAIF*, **11**, 2 (December 2016), 138–156.
- Kurz, G.** and Hanebeck, U. D., Dynamic Surface Reconstruction by Recursive Fusion of Depth and Position Measurements *JAIF*, **9**, 1 (July 2014), 13–26.
- Kurz, G.**, *see* Pfaff, F., *JAIF*, **11**, 2 (December 2016), 206–226.
- Kwan, C.**, *see* Chen, G., *JAIF*, **2**, 1 (July 2007), 35–48.

L

- Le Bihan, N.**, Flamant, J., Manton, J. H., Density Estimation on the Rotation Group Using Diffusive Wavelets, *JAIF*, **11**, 2 (December 2016), 173–185.
- La Cour, B. R.**, *see* Aughenbaugh, J. M., *JAIF*, **5**, 2 (Dec. 2010), 108–127.
- Lang, T.**, *see* Habtemariam, B. K., *JAIF*, **7**, 2 (December 2012), 114–130.
- Lecornu, L.**, *see* Alsun, M., *JAIF*, **7**, 2 (December 2012), 101–113.
- Levchuck, G.**, *see* Han, X., *JAIF*, **7**, 1 (June 2012), 78–96.
- Levedahl, M.**, *see* Areta, J., *JAIF*, **1**, 2 (Dec. 2006), 144–157.
- Li, X. R.**, *see* Jilkov, V. P., *JAIF*, **4**, 1 (July 2009), 27–39.

M

- Manton, J. H.**, *see* Le Bihan, N., *JAIF*, **11**, 2 (December 2016), 173–185.
- Manton, J. H.**, *see* Said, S., *JAIF*, **11**, 2 (December 2016), 227–249.
- Marano, S.**, *see* Georgescu, R., *JAIF*, **7**, 1 (June 2012), 61–77.

Marković, I., Bukal, M., Česić, J., Petrović, I., Multitarget Tracking with the von Mises-Fisher Filter and Probabilistic Data Association, *JAIF*, **11**, 2 (December 2016), 157–172.

Martin, A., Osswald, C., Dezert, J., Smarandache, F., General Combination Rules for Qualitative and Quantitative Beliefs, *JAIF*, **3**, 2 (Dec. 2008), 67–89.

Matta, V., see Georgescu, R., *JAIF*, **7**, 1 (June 2012), 61–77.

Miceli, P. A., see Blair, W. D., *JAIF*, **7**, 1 (June 2012), 28–45.

Milgram, M., see Belaroussi, R., *JAIF*, **1**, 1 (July 2006), 35–51.

Moyer, S., see Howell, C., *JAIF*, **8**, 2 (December 2013), 133–142.

Mullen, T., see Avasarala, V., *JAIF*, **4**, 1 (July 2009), 52–71.

Myrhaug, H., see Kaliciak, L., *JAIF*, **10**, 2 (December 2015), 183–198.

N

Nagi, R., see Sambhoos, K., *JAIF*, **3**, 2 (Dec. 2008), 90–106.

Nadarajah, N., see Tharmarasa, R., *JAIF*, **7**, 1 (June 2012), 46–60.

Ng, G. W., see Foo, P. H., *JAIF*, **4**, 1 (July 2009), 3–26.

Ng, K. H., see Foo, P. H., *JAIF*, **4**, 1 (July 2009), 3–26.

Ng, G. W., see Foo, P. H., *JAIF*, **8**, 1 (July 2013), 33–72.

Nitzan, E., Routtenberg, T., Tabrikian, J., Stochastic Filtering Using Periodic Cost Functions, *JAIF*, **11**, 2 (December 2016), 123–137.

O

Oispuu, M., see Steffes, C., *JAIF*, **11**, 2 (December 2016), 250–261.

Osborne, R. W., III, Bar-Shalom, Y., George, J., Kaplan, L., Statistical Efficiency of Target Localization from Angle and Shockwave Measurements, *JAIF*, **9**, 2 (December 2014), 75–89.

Osborne, R. W., III, Bar-Shalom, Y., Willett, P., Fusion of Asynchronous Passive Measurements, *JAIF*, **10**, 2 (December 2015), 199–210.

Osborne, R. W., III, Bar-Shalom, Y., Willett, P., Track-to-Track Association with Augmented State, *JAIF*, **7**, 1 (June 2012), 3–15.

Osborne, R. W., III, see Belfadel, D., *JAIF*, **9**, 2 (December 2014), 59–74.

Osborne, R. W., III, see Belfadel, D., *JAIF*, **10**, 2 (December 2015), 101–112.

Osborne, R. W., III, see Crouse, D. F., *JAIF*, **8**, 1 (July 2013), 73–89.

Osswald, C., see Martin, A., *JAIF*, **3**, 2 (Dec. 2008), 67–89.

P

Panakkal, V. P. and Velmurugan, R., An improved measurement model for target tracking under measurement origin uncertainty *JAIF*, **10**, 2 (December 2015), 143–162.

Pander, M., see Steinbring, J., *JAIF*, **11**, 1 (June 2016), 71–90.

Papageorgiou, D. J. and Holender, M., Track-to-Track Association and Ambiguity Management in the Presence of Sensor Bias *JAIF*, **6**, 2 (December 2011), 77–100.

Paradis, S., see Benaskeur, A. R., *JAIF*, **2**, 2 (Dec. 2007), 99–112.

Pattipati, K. R., see An, W., *JAIF*, **3**, 1 (July 2008), 33–49.

Pattipati, K. R., see Areta, J., *JAIF*, **1**, 2 (Dec. 2006), 144–157.

Pattipati, K. R., see Areta, J., *JAIF*, **4**, 1 (July 2009), 40–51.

Pattipati, K., see Crouse, D. F., *JAIF*, **8**, 1 (July 2013), 73–89.

Pattipati, K., see Han, X., *JAIF*, **7**, 1 (June 2012), 78–96.

Pattipati, K. R., see Tian, X., *JAIF*, **5**, 1 (July 2010), 3–17.

Peel, L., Estimating Network Parameters for Selecting Community Detection Algorithms *JAIF*, **6**, 2 (December 2011), 119–130.

Petrović, I., see Marković, I., *JAIF*, **11**, 2 (December 2016), 157–172.

Pfaff, F., Kurz, G., Hanebeck, U. D., Multivariate Angular Filtering Using Fourier Series, *JAIF*, **11**, 2 (December 2016), 206–226.

Piché, R., see Raitoharju, M., *JAIF*, **11**, 1 (June 2016), 3–14.

Pignol, A.-C., see Jauffret, C., *JAIF*, **6**, 1 (June 2011), 24–38.

Pillon, D., see Jauffret, C., *JAIF*, **6**, 1 (June 2011), 24–38.

Powell, G. M., see Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.

Prevost, L., see Belaroussi, R., *JAIF*, **1**, 1 (July 2006), 35–51.

Punithakumar, K., see Habtemariam, B. K., *JAIF*, **7**, 2 (December 2012), 114–130.

R

Raitoharju, M., Piché, R., Ala-Luhtala, J., Ali-Löyty, S., Partitioned Update Kalman Filter, *JAIF*, **11**, 1 (June 2016), 3–14.

Ravindra, V. C., Bar-Shalom, Y., and Damarla, T., Feature-Aided Tracking of Ground Vehicles using Passive Acoustic Sensor Arrays, *JAIF*, **5**, 2 (Dec. 2010), 88–107.

Rhéaume, F., see Benaskeur, A. R., *JAIF*, **2**, 2 (Dec. 2007), 99–112.

Reindl, L., see Hamp, Q., *JAIF*, **7**, 2 (December 2012), 131–140.

Rickard, T., see Sambhoos, K., *JAIF*, **3**, 2 (Dec. 2008), 90–106.

Rickard, T., see Sudit, M., *JAIF*, **2**, 1 (July 2007), 3–21.

Rodningsby, A., Bar-Shalom, Y., Hallingstad, O., and Glattetre, J., Multitarget Multisensor Tracking in the Presence of Wakes, *JAIF*, **4**, 2 (Dec. 2009), 117–145.

Rogova, G. L., Bursik, M. I., and Hanson-Hedgecock, S., Intelligent System for Interpreting the Pattern of Volcanic Eruptions, *JAIF*, **3**, 2 (Dec. 2008), 118–128.

Romeo, K., Bar-Shalom, Y., Willett, P., Fusion of Multipath Data with ML-PMHT for Very Low SNR Track Detection in an OTHR, *JAIF*, **10**, 2 (December 2015), 113–124.

Roth, M., Hendeby, G., Gustafsson, F., Nonlinear Kalman Filters Explained: A Tutorial on Moment Computations and Sigma Point Methods, *JAIF*, **11**, 1 (June 2016), 47–70.

Rothrock, R., see Areta, J., *JAIF*, **2**, 2 (Dec. 2007), 113–127.

Routtenberg, T., see Eyal Nitzan, E., *JAIF*, **11**, 2 (December 2016), 123–137.

Ruspini, E. H., see Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.

S

Said, S. and Manton, J. H., Particle Filtering with Observations in a Manifold: A Proof of Convergence and Two Worked Examples *JAIF*, **11**, 2 (December 2016), 227–249.

Salerno, E., Adurthi, N., Singh, T., Singla, P., Bubalo, A., Cornacchia, M., Alford, M., Jones, E., Road Network Identification by means of the Hough Transform with Uncertainty Analysis, *JAIF*, **10**, 1 (June 2015), 58–72.

Salerno, J., see Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.

Särkkä, S., Hartikainen, J., Svensson, L., Sandblom, F., On the relation between Gaussian process quadratures and sigma-point methods, *JAIF*, **11**, 1 (June 2016), 31–46.

Särkkä, S., see Kokkala, J., *JAIF*, **11**, 1 (June 2016), 15–30.

Sambhoos, K., Nagi, R., Sudit, M., and Rickard, T., Hierarchical Higher Level Data Fusion using Fuzzy Hamming and Hypercube Clustering, *JAIF*, **3**, 2 (Dec. 2008), 90–106.

Sandblom, F., see Särkkä, S., *JAIF*, **11**, 1 (June 2016), 31–46.

Schlosser, M. S. and Kroschel, K., Performance Analysis of Decentralized Kalman Filters under Communication Constraints *JAIF*, **2**, 2 (Dec. 2007), 65–76.

Schneider, M. K., see Washburn, R. B., *JAIF*, **3**, 1 (July 2008), 3–13.

Schrag, R. C., Takikawa, M., Goger, P., and Eilbert, J., Performance Evaluation for Automated Threat Detection, *JAIF*, **2**, 2 (Dec. 2007), 77–98.

Scott, P. D., see Terejanu, G., *JAIF*, **5**, 2 (Dec. 2010), 73–87.

Shen, D., see Chen, G., *JAIF*, **2**, 1 (July 2007), 35–48.

Siegrwart, R. Y., see Kurz, G., *JAIF*, **11**, 2 (December 2016), 138–156.

Šimandl, M., see Dunfk, J., *JAIF*, **11**, 1 (June 2016), 91–109.

Singh, S., see An, W., *JAIF*, **3**, 1 (July 2008), 33–49.

Singh, T., see George, J., *JAIF*, **6**, 1 (June 2011), 39–56.

Singh, T., see Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.

Singh, T., see Terejanu, G., *JAIF*, **5**, 2 (Dec. 2010), 73–87.

Singla, P., see Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.

Singla, P., see Terejanu, G., *JAIF*, **5**, 2 (Dec. 2010), 73–87.

Smarandache, F., see Martin, A., *JAIF*, **3**, 2 (Dec. 2008), 67–89.

Sndarö, L., see Biermann, J., *JAIF*, **8**, 2 (December 2013), 101–118.

Solaiman, B., see Alsun, M., *JAIF*, **7**, 2 (December 2012), 101–113.

Solin, A., see Kokkala, J., *JAIF*, **11**, 1 (June 2016), 15–30.

Song, D., see Kaliciak, L., *JAIF*, **10**, 2 (December 2015), 183–198.

Song, X., Willett, P., Zhou, S., Bearings-Only Localization with NLOS Reflected AoAs, *JAIF*, **8**, 1 (July 2013), 3–14.

Song, X., Willett, P., Zhou, S., Posterior Cramér-Rao Bounds for Doppler Biased Distributed Tracking, *JAIF*, **7**, 1 (June 2012), 16–27.

Steinbring, J. and Hanebeck, U. D., LRKF Revisited: The Smart Sampling Kalman Filter (S² KF) *JAIF*, **9**, 2 (December 2014), 106–123.

Steffes, C. and Oispuu, M., Direct Position Determination for TDOA-based Single Sensor Localization *JAIF*, **11**, 2 (December 2016), 250–261.

Steinbring, J., Pander, M., Hanebeck, U. D., The Smart Sampling Kalman Filter with Symmetric Samples, *JAIF*, **11**, 1 (June 2016), 71–90.

Stone, L. D. and Anderson, S. L., Standard Bayesian Approach to Quantized Measurements and Imprecise Likelihoods *JAIF*, **10**, 1 (June 2015), 3–12.

Stotz, A., see Sudit, M., *JAIF*, **2**, 1 (July 2007), 3–21.

Straka, O., see Dunfk, J., *JAIF*, **11**, 1 (June 2016), 91–109.

Streit, R., A Technique for Deriving Multitarget Intensity Filters Using Ordinary Derivatives *JAIF*, **9**, 1 (July 2014), 3–12.

Streit, R., The Probability Generating Functional for Finite Point Processes, and Its Application to the Comparison of PHD and Intensity Filters *JAIF*, **8**, 2 (December 2013), 119–132.

Sudit, M., Holender, M., Stotz, A., Rickard, T., and Yager, R., INFERD and Entropy for Situational Awareness, *JAIF*, **2**, 1 (July 2007), 3–21.

Sudit, M., *see* Sambhoos, K., *JAIF*, **3**, 2 (Dec. 2008), 90–106.
Svensson, L., *see* Särkkä, S., *JAIF*, **11**, 1 (June 2016), 31–46.

T

Tabrikian, J., *see* Eyal Nitzan, E., *JAIF*, **11**, 2 (December 2016), 123–137.
Takikawa, M., *see* Schrag, R. C., *JAIF*, **2**, 2 (Dec. 2007), 77–98.
Terejanu, G., Singla, P., Singh, T., and Scott, P. D., A Decision-Centric Framework for Density Forecasting, *JAIF*, **5**, 2 (Dec. 2010), 73–87.
Tharmarasa, R., *see* Habtemariam, B. K., *JAIF*, **7**, 2 (December 2012), 114–130.
Tharmarasa, R., Kirubarajan, T., Nadarajah, N., Bar-Shalom, Y., Thayaparan, T., Profile-Free Launch Point Estimation for Ballistic Targets using Passive Sensors, *JAIF*, **7**, 1 (June 2012), 46–60.
Thayaparan, T., *see* Tharmarasa, R., *JAIF*, **7**, 1 (June 2012), 46–60.
Tian, X., Bar-Shalom, Y., and Pattipati, K. R., Multi-step Look-Ahead Policy for Autonomous Cooperative Surveillance by UAVs in Hostile Environments, *JAIF*, **5**, 1 (July 2010), 3–17.
Tian, X. and Bar-Shalom, Y., Algorithms for Asynchronous Track-to-Track Fusion *JAIF*, **5**, 2 (Dec. 2010), 128–138.
Tian, X., *see* Yuan, T., *JAIF*, **6**, 2 (December 2011), 131–149.
Tian, X. and Bar-Shalom, Y., Track-to-Track Fusion Configurations and Association in a Sliding Window *JAIF*, **4**, 2 (Dec. 2009), 146–164.
Tu, F., *see* Han, X., *JAIF*, **7**, 1 (June 2012), 78–96.

V

Valin, P., Djiknavorian, P., and Bossé, É., A Pragmatic Approach for the use of Dempster-Shafer Theory in Fusing Realistic Sensor Data, *JAIF*, **5**, 1 (July 2010), 32–40.
Velmurugan, R., *see* Panakkal, V. P., *JAIF*, **10**, 2 (December 2015), 143–162.

W

Washburn, R. B. and Schneider, M. K., Optimal Policies for a Class of Restless Multiarmed Bandit Scheduling Problems with Applications to Sensor Management *JAIF*, **3**, 1 (July 2008), 3–13.
Willett, P., *see* Choi, S., *JAIF*, **8**, 2 (December 2013), 143–155.
Willett, P., *see* Choi, S., *JAIF*, **9**, 1 (July 2014), 27–37.
Willett, P., *see* Coraluppi, S., *JAIF*, **5**, 1 (July 2010), 18–31.
Willett, P., *see* Crouse, D. F., *JAIF*, **4**, 2 (Dec. 2009), 93–116.
Willett, P., *see* Crouse, D. F., *JAIF*, **7**, 1 (June 2012), 97–98.
Willett, P., *see* Crouse, D. F., *JAIF*, **8**, 1 (July 2013), 73–89.
Willett, P., *see* Dou, W., *JAIF*, **10**, 2 (December 2015), 163–182.
Willett, P., *see* Erdinc, O., *JAIF*, **2**, 1 (July 2007), 22–34.
Willett, P., *see* Georgescu, R., *JAIF*, **7**, 1 (June 2012), 61–77.
Willett, P., *see* Osborne, R. W., III, *JAIF*, **7**, 1 (June 2012), 3–15.
Willett, P., *see* Osborne, R. W., III, *JAIF*, **10**, 2 (December 2015), 199–210.
Willett, P., *see* Romeo, K., *JAIF*, **10**, 2 (December 2015), 113–124.
Willett, P., *see* Song, X., *JAIF*, **7**, 1 (June 2012), 16–27.
Willett, P., *see* Song, X., *JAIF*, **8**, 1 (July 2013), 3–14.
Wu, J., *see* Jilkov, V. P., *JAIF*, **10**, 1 (June 2015), 73–88.

Y

Yager, R., *see* Sudit, M., *JAIF*, **2**, 1 (July 2007), 3–21.
Yang, C. and Blasch, E., Fusion of Tracks with Road Constraints *JAIF*, **3**, 1 (July 2008), 14–32.
Yang, R., *see* Foo, P. H., *JAIF*, **4**, 1 (July 2009), 3–26.
Yu, F., *see* Han, X., *JAIF*, **7**, 1 (June 2012), 78–96.
Yuan, T., Bar-Shalom, Y., Tian, X., Heterogeneous Track-to-Track Fusion, *JAIF*, **6**, 2 (December 2011), 131–149.

Z

Zea, A., *see* Faion, F., *JAIF*, **10**, 1 (June 2015), 13–30.
Zhang, S. and Bar-Shalom, Y., Practical Data Association for Passive Sensors in 3D *JAIF*, **9**, 1 (July 2014), 38–46.

Zhang, S., Bar-Shalom, Y., Watson, G., Tracking with Multisensor Out-of-Sequence Measurements with Residual Biases, *JAIF*, **6**, 1 (June 2011), 3–23.
Zheng, L., *see* Crouse, D. F., *JAIF*, **7**, 1 (June 2012), 97–98.
Zheng, Y. and Blasch, E., An Exploration of the Impacts of Three Factors in Multimodal Biometric Score Fusion: Score Modality, Recognition Method, and Fusion Process *JAIF*, **9**, 2 (December 2014), 124–135.
Zhou, S., *see* Choi, S., *JAIF*, **9**, 1 (July 2014), 27–37.
Zhou, S., *see* Song, X., *JAIF*, **7**, 1 (June 2012), 16–27.
Zhou, S., *see* Song, X., *JAIF*, **8**, 1 (July 2013), 3–14.

SUBJECT INDEX

0–1 nonlinear optimization

Papageorgiou, D. J., +, *JAIF*, **6**, 2 (December 2011), 77–100.

A

acoustic localization

Osborne, R. W., III, +, *JAIF*, **9**, 2 (December 2014), 75–89.

adaptive encoding

Habtemariam, B. K., +, *JAIF*, **7**, 2 (December 2012), 114–130.

Adaptive Gaussian Mixture Models

Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.

adaptive model selection

Tharmarasa, R., +, *JAIF*, **7**, 1 (June 2012), 46–60.

air defense

Johansson, F., +, *JAIF*, **6**, 2 (December 2011), 187–199.

alerting

Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.

algorithm selection

Peel, L., *JAIF*, **6**, 2 (December 2011), 119–130.

ambiguity management

Papageorgiou, D. J., +, *JAIF*, **6**, 2 (December 2011), 77–100.

angular quantities

Kurz, G., +, *JAIF*, **9**, 2 (December 2014), 90–105.

anomaly detection

George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.

approximate crosscorrelation

Bar-Shalom, Y., +, *JAIF*, **3**, 2 (Dec. 2008), 107–117.

assignment

Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.

Association ambiguity

Choi, S., +, *JAIF*, **9**, 1 (July 2014), 27–37.

association benchmark

Areta, J., +, *JAIF*, **1**, 2 (Dec. 2006), 144–157.

association

Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.

Bar-Shalom, Y., +, *JAIF*, **2**, 1 (July 2007), 49–59.

Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.

Hamp, Q., +, *JAIF*, **7**, 2 (December 2012), 131–140.

Tian, X., +, *JAIF*, **4**, 2 (Dec. 2009), 146–164.

asynchronous

Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.

Asynchronous Track-to-Track fusion

Tian, X., +, *JAIF*, **5**, 2 (Dec. 2010), 128–138.

augmented state

Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.

Osborne, R. W., III, +, *JAIF*, **7**, 1 (June 2012), 3–15.

automatic target recognition (ATR)

Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.

B

ballistic targets

Tharmarasa, R., +, *JAIF*, **7**, 1 (June 2012), 46–60.

Bayesian

Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.

Bayesian and Reasoning Methods

Arnborg, S., *JAIF*, **1**, 1 (July 2006), 75–90.

Brynielsson, J., +, *JAIF*, **1**, 2 (Dec. 2006), 108–121.

Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.

Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.

Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.

Valin, P., +, *JAIF*, **5**, 1 (July 2010), 32–40.

Washburn, R. B., +, *JAIF*, **3**, 1 (July 2008), 3–13.

Bayesian Data Fusion

Junghans, M., +, *JAIF*, **3**, 1 (July 2008), 50–62.

Bayesian filtering
Blom, H. A. P., +, *J AIF*, **1**, 1 (July 2006), 15–34.

Bayesian filters
Dunfk, J., +, *J AIF*, **11**, 1 (June 2016), 91–109.

Bayesian Game
Brynielsson, J., +, *J AIF*, **1**, 2 (Dec. 2006), 108–121.

Bayesian Networks (BNs)
Junghans, M., +, *J AIF*, **3**, 1 (July 2008), 50–62.

Bayesian network
Chang, K. C., +, *J AIF*, **1**, 2 (Dec. 2006), 95–107.

Bayesian theory
Karlsson, A., +, *J AIF*, **6**, 2 (December 2011), 150–166.

Bayesian tracking
Aughenbaugh, J. M., +, *J AIF*, **5**, 2 (Dec. 2010), 108–127.

bearings-only target motion analysis
Jauffret, C., +, *J AIF*, **6**, 1 (June 2011), 24–38.

benchmark performance
Howell, C., +, *J AIF*, **8**, 2 (December 2013), 133–142.

belief function theory
Martin, A., +, *J AIF*, **3**, 2 (Dec. 2008), 67–89.

bias
Ferry, J. P., *J AIF*, **5**, 1 (July 2010), 41–67.

Bias estimation
Belfadel, D., +, *J AIF*, **9**, 2 (December 2014), 59–74.
Belfadel, D., +, *J AIF*, **10**, 2 (December 2015), 101–112.

bias observability
Belfadel, D., +, *J AIF*, **9**, 2 (December 2014), 59–74.

biased measurement
Zhang, S., +, *J AIF*, **6**, 1 (June 2011), 3–23.

bias-variance trade-off
George, J., +, *J AIF*, **8**, 1 (July 2013), 15–32.

BOMTMA
Jauffret, C., +, *J AIF*, **6**, 1 (June 2011), 24–38.

BOTMA
Jauffret, C., +, *J AIF*, **6**, 1 (June 2011), 24–38.

brake-by-wire
Hoseinnezhad, R., +, *J AIF*, **1**, 1 (July 2006), 52–62.

C

capacities
Arnborg, S., *J AIF*, **1**, 1 (July 2006), 75–90.

cardinality tracking
Coraluppi, S., +, *J AIF*, **7**, 2 (December 2012), 153–164.

centralized fusion
Rodningsby, A., +, *J AIF*, **4**, 2 (Dec. 2009), 117–145.

Characteristic function
Le Bihan, N., +, *J AIF*, **11**, 1 (June 2016), 173–185.

circular data
Kurz, G., +, *J AIF*, **9**, 2 (December 2014), 90–105.

circular statistics
Crouse, D. F., +, *J AIF*, **8**, 1 (July 2013), 73–89.
Kurz, G., +, *J AIF*, **11**, 1 (June 2016), 138–156.

Circular stochastic filtering
Nitzan, E., +, *J AIF*, **11**, 1 (June 2016), 123–137.

classification
Hamp, Q., +, *J AIF*, **7**, 2 (December 2012), 131–140.

Classification, Learning, Data Mining
Jabeur, K., +, *J AIF*, **4**, 2 (Dec. 2009), 75–92.
Martin, A., +, *J AIF*, **3**, 2 (Dec. 2008), 67–89.
Sambhoos, K., +, *J AIF*, **3**, 2 (Dec. 2008), 90–106.

clustering
Sambhoos, K., +, *J AIF*, **3**, 2 (Dec. 2008), 90–106.

clutter suppression
Kahler, B., +, *J AIF*, **6**, 2 (December 2011), 101–118.

collation
Biermann, J., +, *J AIF*, **8**, 2 (December 2013), 101–118.

Combat Power Management
Benaskeur, A. R., +, *J AIF*, **2**, 2 (Dec. 2007), 99–112.

combination
Areta, J., +, *J AIF*, **4**, 1 (July 2009), 40–51.
Belaroussi, R., +, *J AIF*, **1**, 1 (July 2006), 35–51.

combination operators
Karlsson, A., +, *J AIF*, **6**, 2 (December 2011), 150–166.

combinatorial auctions
Avasarala, V., +, *J AIF*, **4**, 1 (July 2009), 52–71.

Command and Control
Brynielsson, J., +, *J AIF*, **1**, 2 (Dec. 2006), 108–121.

communication rate reduction
Schlosser, M. S., +, *J AIF*, **2**, 2 (Dec. 2007), 65–76.

Community detection
Peel, L., *J AIF*, **6**, 2 (December 2011), 119–130.

composite measurement
Osborne, III, R. W., +, *J AIF*, **10**, 2 (December 2015), 199–210.

composite measurements
Belfadel, D., +, *J AIF*, **9**, 2 (December 2014), 59–74.
Belfadel, D., +, *J AIF*, **10**, 2 (December 2015), 101–112.

Con-Tracker
George, J., +, *J AIF*, **6**, 1 (June 2011), 39–56.

connector map
Said, S., +, *J AIF*, **11**, 1 (June 2016), 227–249.

conflict management
Martin, A., +, *J AIF*, **3**, 2 (Dec. 2008), 67–89.

conservative fusion approach
Schlosser, M. S., +, *J AIF*, **2**, 2 (Dec. 2007), 65–76.

Content-based Image Retrieval
Kaliciak, L., +, *J AIF*, **10**, 2 (December 2015), 183–198.

context-aware modeling
George, J., +, *J AIF*, **6**, 1 (June 2011), 39–56.

Continuous time systems
Crouse, D. F., +, *J AIF*, **10**, 1 (June 2015), 31–57.

convergent analysis
Carvalho, R. N., +, *J AIF*, **7**, 2 (December 2012), 141–152.

coordinated turn
Yuan, T., +, *J AIF*, **6**, 2 (December 2011), 131–149.

Cramér-Rao lower bound
Jauffret, C., +, *J AIF*, **6**, 1 (June 2011), 24–38.

credal sets
Karlsson, A., +, *J AIF*, **6**, 2 (December 2011), 150–166.

CRLB
Belfadel, D., +, *J AIF*, **9**, 2 (December 2014), 59–74.
Belfadel, D., +, *J AIF*, **10**, 2 (December 2015), 101–112.
Osborne, R. W., III, +, *J AIF*, **9**, 2 (December 2014), 75–89.
Osborne, III, R. W., +, *J AIF*, **10**, 2 (December 2015), 199–210.

D

data association
Areta, J., +, *J AIF*, **1**, 2 (Dec. 2006), 144–157.
Osborne, R. W., III, +, *J AIF*, **7**, 1 (June 2012), 3–15.
Rodningsby, A., +, *J AIF*, **4**, 2 (Dec. 2009), 117–145.
Zhang, S., +, *J AIF*, **9**, 1 (July 2014), 38–46.

data fusion
Chen, G., +, *J AIF*, **2**, 1 (July 2007), 35–48.
Foo, P. H., +, *J AIF*, **8**, 1 (July 2013), 33–72.
Papageorgiou, D. J., +, *J AIF*, **6**, 2 (December 2011), 77–100.

Data Qualification
Junghans, M., +, *J AIF*, **3**, 1 (July 2008), 50–62.

Decentralized Kalman Filter (DKF)
Schlosser, M. S., +, *J AIF*, **2**, 2 (Dec. 2007), 65–76.

decision making
Terejanu, G., +, *J AIF*, **5**, 2 (Dec. 2010), 73–87.

Decision-Level Fusion (DLF)
Kahler, B., +, *J AIF*, **6**, 2 (December 2011), 101–118.

Dempster-Shafer
Stone, L. D., +, *J AIF*, **10**, 1 (June 2015), 3–12.
Valin, P., +, *J AIF*, **5**, 1 (July 2010), 32–40.

Density Forecasting
Terejanu, G., +, *J AIF*, **5**, 2 (Dec. 2010), 73–87.

depth camera
Kurz, G., +, *J AIF*, **9**, 1 (July 2014), 13–26.

decision fusion
Rogova, G. L., +, *J AIF*, **3**, 2 (Dec. 2008), 118–128.

descriptor system
Blom, H. A. P., +, *J AIF*, **1**, 1 (July 2006), 15–34.

detection
Davey, S. J., *J AIF*, **6**, 2 (December 2011), 167–186.
Jilkov, V. P., +, *J AIF*, **4**, 1 (July 2009), 27–39.

deterministic sampling
Kurz, G., +, *J AIF*, **11**, 1 (June 2016), 138–156.

differentiable manifold
Said, S., +, *J AIF*, **11**, 1 (June 2016), 227–249.

Differentiation

Streit, R., +, *JAIF*, **9**, 1 (July 2014), 3–12.

diffusion process

Said, S., +, *JAIF*, **11**, 1 (June 2016), 227–249.

Diffusive wavelets

Le Bihan, N., +, *JAIF*, **11**, 1 (June 2016), 173–185.

Digital audio/video broadcasting

Choi, S., +, *JAIF*, **9**, 1 (July 2014), 27–37.

Dirac Mixtures

Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.

Kurz, G., +, *JAIF*, **11**, 1 (June 2016), 138–156.

direction of arrival (DoA) estimation

Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.

direction-of-arrival (DOA) tracking

Nitzan, E., +, *JAIF*, **11**, 1 (June 2016), 123–137.

directional statistics

Crouse, D. F., +, *JAIF*, **8**, 1 (July 2013), 73–89.

Kurz, G., +, *JAIF*, **9**, 2 (December 2014), 90–105.

Pfaff, F., +, *JAIF*, **11**, 1 (June 2016), 206–226.

discounting operators

Karlsson, A., +, *JAIF*, **6**, 2 (December 2011), 150–166.

Displaced Phase Center Antenna (DPCA)

Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.

distributed detection

Coraluppi, S., +, *JAIF*, **5**, 1 (July 2010), 18–31.

distributed inference

Choi, S., +, *JAIF*, **8**, 2 (December 2013), 143–155.

distributed tracking

Habtariam, B. K., +, *JAIF*, **7**, 2 (December 2012), 114–130.

Divided Differences

Michael Roth, +, *JAIF*, **11**, 1 (June 2016), 47–70.

DS-structures

Arnborg, S., *JAIF*, **1**, 1 (July 2006), 75–90.

DSmT

Martin, A., +, *JAIF*, **3**, 2 (Dec. 2008), 67–89.

dynamic programming

Washburn, R. B., +, *JAIF*, **3**, 1 (July 2008), 3–13.

E

early fusion

Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.

EM Algorithm

Crouse, D. F., +, *JAIF*, **4**, 2 (Dec. 2009), 93–116.

entropy

Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.

estimator efficiency

Osborne, R. W., III, +, *JAIF*, **9**, 2 (December 2014), 75–89.

Expected Loss

Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.

experiment design

Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.

experimental results

George, J., +, *JAIF*, **8**, 1 (July 2013), 15–32.

Extended Object Tracking

Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.

F

false measurements

Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.

false diagnosis

Choi, S., +, *JAIF*, **8**, 2 (December 2013), 143–155.

features

Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.

feature extraction

Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.

feature-aided tracking

Coraluppi, S., +, *JAIF*, **6**, 1 (June 2011), 57–67.

FIM

Osborne, R. W., III, +, *JAIF*, **9**, 2 (December 2014), 75–89.

Finite point processes, Multitarget tracking

Streit, R., +, *JAIF*, **9**, 1 (July 2014), 3–12.

Fire Control

Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.

flight profile

Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.

Fourier series

Pfaff, F., +, *JAIF*, **11**, 1 (June 2016), 206–226.

Fourier-based stochastic filtering via root-finding (FB-SFRF)

Nitzan, E., +, *JAIF*, **11**, 1 (June 2016), 123–137.

fuse-before-track

Coraluppi, S., +, *JAIF*, **5**, 1 (July 2010), 18–31.

fusion

Crouse, D. F., +, *JAIF*, **4**, 2 (Dec. 2009), 93–116.

Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.

Jilkov, V. P., +, *JAIF*, **4**, 1 (July 2009), 27–39.

Osborne, R. W., III, +, *JAIF*, **9**, 2 (December 2014), 75–89.

Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.

Tian, X., +, *JAIF*, **4**, 2 (Dec. 2009), 146–164.

Fusion Applications

Areta, J., +, *JAIF*, **1**, 2 (Dec. 2006), 144–157.

Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.

Chen, G., +, *JAIF*, **2**, 1 (July 2007), 35–48.

Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.

Hoseinnezhad, R., +, *JAIF*, **1**, 1 (July 2006), 52–62.

Junghans, M., +, *JAIF*, **3**, 1 (July 2008), 50–62.

Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.

Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.

Yang, C., +, *JAIF*, **3**, 1 (July 2008), 14–32.

Fusion Architectures and Management Issues

Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.

Avasarala, V., +, *JAIF*, **4**, 1 (July 2009), 52–71.

Blasch, E., +, *JAIF*, **1**, 2 (Dec. 2006), 122–143.

Chang, K. C., +, *JAIF*, **1**, 2 (Dec. 2006), 95–107.

Coraluppi, S., +, *JAIF*, **5**, 1 (July 2010), 18–31.

Schlosser, M. S., +, *JAIF*, **2**, 2 (Dec. 2007), 65–76.

Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.

Tian, X., +, *JAIF*, **5**, 1 (July 2010), 3–17.

fusion of clustering results

Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.

fuzzy

Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.

fuzzy hamming distance

Sambhoos, K., +, *JAIF*, **3**, 2 (Dec. 2008), 90–106.

fuzzy inference

Belaroussi, R., +, *JAIF*, **1**, 1 (July 2006), 35–51.

fuzzy logic

Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.

fuzzy rule-base

Hoseinnezhad, R., +, *JAIF*, **1**, 1 (July 2006), 52–62.

G

game theory

Brynielsson, J., +, *JAIF*, **1**, 2 (Dec. 2006), 108–121.

Chen, G., +, *JAIF*, **2**, 1 (July 2007), 35–48.

Gauss-Newton method

George, J., +, *JAIF*, **8**, 1 (July 2013), 15–32.

genetic algorithms

Avasarala, V., +, *JAIF*, **4**, 1 (July 2009), 52–71.

Jabeur, K., +, *JAIF*, **4**, 2 (Dec. 2009), 75–92.

Geodesy

Crouse, D. F., +, *JAIF*, **10**, 1 (June 2015), 31–57.

geological data

Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.

global

Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.

Graph Matching

Han, X., +, *JAIF*, **7**, 1 (June 2012), 78–96.

grid-based filtering

Aughenbaugh, J. M., +, *JAIF*, **5**, 2 (Dec. 2010), 108–127.

GMTI

Salerno, E., +, *JAIF*, **10**, 1 (June 2015), 58–72.

gunfire detection system

George, J., +, *JAIF*, **8**, 1 (July 2013), 15–32.

H

heterogeneous track-to-track fusion

Yuan, T., +, *JAIF*, **6**, 2 (December 2011), 131–149.

hidden Markov model

An, W., +, *JAIF*, **3**, 1 (July 2008), 33–49.

hidden Markov model (HMM)

Zheng, Y., +, *JAIF*, **9**, 2 (December 2014), 124–135.

hierarchical entity aggregation

Chen, G., +, *JAIF*, 2, 1 (July 2007), 35–48.

High Level Data Fusion

Sambhoos, K., +, *JAIF*, 3, 2 (Dec. 2008), 90–106.

High Range Resolution (HRR)

Kahler, B., +, *JAIF*, 6, 2 (December 2011), 101–118.

histogram PMHT

Davey, S. J., *JAIF*, 6, 2 (December 2011), 167–186.

Hough transform

Belaroussi, R., +, *JAIF*, 1, 1 (July 2006), 35–51.

Salerno, E., +, *JAIF*, 10, 1 (June 2015), 58–72.

human cognition

Foo, P. H., +, *JAIF*, 4, 1 (July 2009), 3–26.

Human-Fusion Interaction

Blasch, E., +, *JAIF*, 1, 2 (Dec. 2006), 122–143.

human identification

Zheng, Y., +, *JAIF*, 9, 2 (December 2014), 124–135.

Human-Machine Interface

Blasch, E., +, *JAIF*, 1, 2 (Dec. 2006), 122–143.

hybrid relevance feedback

Kaliciak, L., +, *JAIF*, 10, 2 (December 2015), 183–198.

hypercube distance

Sambhoos, K., +, *JAIF*, 3, 2 (Dec. 2008), 90–106.

hypotheses

Areta, J., +, *JAIF*, 4, 1 (July 2009), 40–51.

hypothesis generator

George, J., +, *JAIF*, 6, 1 (June 2011), 39–56.

hypothesis management

Coraluppi, S., +, *JAIF*, 6, 1 (June 2011), 57–67.

I

Image Fusion

Belaroussi, R., +, *JAIF*, 1, 1 (July 2006), 35–51.

image fusion performance assessment

Howell, C., +, *JAIF*, 8, 2 (December 2013), 133–142.

image fusion task performance

Howell, C., +, *JAIF*, 8, 2 (December 2013), 133–142.

imetrics

Schrag, R. C., +, *JAIF*, 2, 2 (Dec. 2007), 77–98.

imprecise

Stone, L. D., +, *JAIF*, 10, 1 (June 2015), 3–12.

imprecise probability

Arnborg, S., *JAIF*, 1, 1 (July 2006), 75–90.

Karlsson, A., +, *JAIF*, 6, 2 (December 2011), 150–166.

index rules

Washburn, R. B., +, *JAIF*, 3, 1 (July 2008), 3–13.

indicator

Biermann, J., +, *JAIF*, 8, 2 (December 2013), 101–118.

INFERD

Sudit, M., +, *JAIF*, 2, 1 (July 2007), 3–21.

Influence Diagram

Brynielsson, J., +, *JAIF*, 1, 2 (Dec. 2006), 108–121.

Information and data fusion

Kaliciak, L., +, *JAIF*, 10, 2 (December 2015), 183–198.

information extraction

Biermann, J., +, *JAIF*, 8, 2 (December 2013), 101–118.

information fusion

Bar-Shalom, Y., +, *JAIF*, 1, 1 (July 2006), 3–14.

Blasch, E., +, *JAIF*, 1, 2 (Dec. 2006), 122–143.

Foo, P. H., +, *JAIF*, 8, 1 (July 2013), 33–72.

Jabeur, K., +, *JAIF*, 4, 2 (Dec. 2009), 75–92.

Sudit, M., +, *JAIF*, 2, 1 (July 2007), 3–21.

information gain heuristic

An, W., +, *JAIF*, 3, 1 (July 2008), 33–49.

information heuristics

Choi, S., +, *JAIF*, 8, 2 (December 2013), 143–155.

Information Retrieval

Kaliciak, L., +, *JAIF*, 10, 2 (December 2015), 183–198.

Integration Rules

Roth, M., +, *JAIF*, 11, 1 (June 2016), 47–70.

intelligence

Biermann, J., +, *JAIF*, 8, 2 (December 2013), 101–118.

intent inference

Foo, P. H., +, *JAIF*, 4, 1 (July 2009), 3–26.

intensity filter

Streit, R., *JAIF*, 8, 2 (December 2013), 119–132.

interaction networks

Peel, L., *JAIF*, 6, 2 (December 2011), 119–130.

Interpolation

Roth, M., +, *JAIF*, 11, 1 (June 2016), 47–70.

IPDA

Chakravorty, R., +, *JAIF*, 1, 1 (July 2006), 63–74.

Iterative solution

Yang, C., +, *JAIF*, 3, 1 (July 2008), 14–32.

K

Kalman filter

George, J., +, *JAIF*, 6, 1 (June 2011), 39–56.

Roth, M., +, *JAIF*, 11, 1 (June 2016), 47–70.

Kernel estimators

Le Bihan, N., +, *JAIF*, 11, 1 (June 2016), 173–185.

Knowledge Representation

Blasch, E., +, *JAIF*, 1, 2 (Dec. 2006), 122–143.

L

L2/L3 fusion

George, J., +, *JAIF*, 6, 1 (June 2011), 39–56.

Lagrangian multiplier

Yang, C., +, *JAIF*, 3, 1 (July 2008), 14–32.

late fusion

Kaliciak, L., +, *JAIF*, 10, 2 (December 2015), 183–198.

launch point estimation

Tharmarasa, R., +, *JAIF*, 7, 1 (June 2012), 46–60.

LCD

Steinbring, J., +, *JAIF*, 9, 2 (December 2014), 106–123.

least squares

Song, X., +, *JAIF*, 8, 1 (July 2013), 3–14.

least squares estimation

Tharmarasa, R., +, *JAIF*, 7, 1 (June 2012), 46–60.

legacy sensors

Bar-Shalom, Y., +, *JAIF*, 3, 2 (Dec. 2008), 107–117.

level 2 fusion

Chang, K. C., +, *JAIF*, 1, 2 (Dec. 2006), 95–107.

level 4 data fusion

Avasarala, V., +, *JAIF*, 4, 1 (July 2009), 52–71.

likelihood

Arnborg, S., *JAIF*, 1, 1 (July 2006), 75–90.

Stone, L. D., +, *JAIF*, 10, 1 (June 2015), 3–12.

likelihood sampling

Aughenbaugh, J. M., +, *JAIF*, 5, 2 (Dec. 2010), 108–127.

linear minimum mean square error

Yuan, T., +, *JAIF*, 6, 2 (December 2011), 131–149.

link-analysis

Biermann, J., +, *JAIF*, 8, 2 (December 2013), 101–118.

localization

Song, X., +, *JAIF*, 8, 1 (July 2013), 3–14.

LOS

Osborne, III, R. W., +, *JAIF*, 10, 2 (December 2015), 199–210.

LRKF

Steinbring, J., +, *JAIF*, 9, 2 (December 2014), 106–123.

M

m-Best Soft Assignment Algorithm

Han, X., +, *JAIF*, 7, 1 (June 2012), 78–96.

Mahler/Fixs?n rule

Arnborg, S., *JAIF*, 1, 1 (July 2006), 75–90.

Maneuvering target

Jauffret, C., +, *JAIF*, 6, 1 (June 2011), 24–38.

maximum likelihood

Belfadel, D., +, *JAIF*, 10, 2 (December 2015), 101–112.

Belfadel, D., +, *JAIF*, 9, 2 (December 2014), 59–74.

Osborne, III, R. W., +, *JAIF*, 10, 2 (December 2015), 199–210.

Song, X., +, *JAIF*, 8, 1 (July 2013), 3–14.

maximum likelihood estimate

Jauffret, C., +, *JAIF*, 6, 1 (June 2011), 24–38.

maximum likelihood estimation

Crouse, D. F., +, *JAIF*, 8, 1 (July 2013), 73–89.

George, J., +, *JAIF*, 8, 1 (July 2013), 15–32.

maximum likelihood fusion

Yuan, T., +, *JAIF*, 6, 2 (December 2011), 131–149.

Mean Integrated Square Error (MISE)

Le Bihan, N., +, *JAIF*, **11**, 1 (June 2016), 173–185.

measurement fusion

Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.

measurement origin uncertainty

Panakkal, V. P., +, *JAIF*, **10**, 2 (December 2015), 143–162.

Medical Decision Support Systems

Alsun, M., +, *JAIF*, **7**, 2 (December 2012), 101–113.

Medical Reasoning

Alsun, M., +, *JAIF*, **7**, 2 (December 2012), 101–113.

MHT

Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.

military application

Jabeur, K., +, *JAIF*, **4**, 2 (Dec. 2009), 75–92.

MIMO radar

Song, X., +, *JAIF*, **7**, 1 (June 2012), 16–27.

minimum variance distortionless response (MVDR)

Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.

misassociation

Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.

Mixture of densities

Le Bihan, N., +, *JAIF*, **11**, 1 (June 2016), 173–185.

ML-PMHT

Romeo, K., +, *JAIF*, **10**, 2 (December 2015), 113–124.

model-based classifier

Belaroussi, R., +, *JAIF*, **1**, 1 (July 2006), 35–51.

modelling

Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.

Moment Computation Problem

Roth, M., +, *JAIF*, **11**, 1 (June 2016), 47–70.

moment matching

Kurz, G., +, *JAIF*, **11**, 1 (June 2016), 138–156.

moving target identification

Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.

multi-channel signal subspace (MSS)

Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.

multi-criteria classification

Jabeur, K., +, *JAIF*, **4**, 2 (Dec. 2009), 75–92.

multi-hypothesis tracking

Coraluppi, S., +, *JAIF*, **5**, 1 (July 2010), 18–31.

Coraluppi, S., +, *JAIF*, **6**, 1 (June 2011), 57–67.

multi-sensor multi-target tracking

Coraluppi, S., +, *JAIF*, **6**, 1 (June 2011), 57–67.

multi-sensor target tracking

Schlosser, M. S., +, *JAIF*, **2**, 2 (Dec. 2007), 65–76.

multitarget tracking

Marković, I., +, *JAIF*, **11**, 1 (June 2016), 157–172.

multidimensional assignment

Areta, J., +, *JAIF*, **1**, 2 (Dec. 2006), 144–157.

Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.

Multimedia Retrieval

Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.

Multimodal biometric score fusion

Zheng, Y., +, *JAIF*, **9**, 2 (December 2014), 124–135.

multiple dimension assignment (MDA)

Zhang, S., +, *JAIF*, **9**, 1 (July 2014), 38–46.

multiobjective optimization

Jilkov, V. P., +, *JAIF*, **4**, 1 (July 2009), 27–39.

multipath

Romeo, K., +, *JAIF*, **10**, 2 (December 2015), 113–124.

multiple-hypothesis tracking

Coraluppi, S., +, *JAIF*, **7**, 2 (December 2012), 153–164.

Multiple-Model Adaptive Estimator

George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.

multisensor tracking

Bar-Shalom, Y., +, *JAIF*, **1**, 1 (July 2006), 3–14.

Blair, W. D., +, *JAIF*, **7**, 1 (June 2012), 28–45.

Yuan, T., +, *JAIF*, **6**, 2 (December 2011), 131–149.

multispectral face recognition

Zheng, Y., +, *JAIF*, **9**, 2 (December 2014), 124–135.

multistatic

Erdinc, O., +, *JAIF*, **2**, 1 (July 2007), 22–34.

multitarget multisensor

Rodningsby, A., +, *JAIF*, **4**, 2 (Dec. 2009), 117–145.

Multitarget Multisensor Tracking

Georgescu, R., +, *JAIF*, **7**, 1 (June 2012), 61–77.

multitarget tracking filters

Streit, R., *JAIF*, **8**, 2 (December 2013), 119–132.

multi-sensor tracking

Coraluppi, S., +, *JAIF*, **5**, 1 (July 2010), 18–31.

multivariate filtering

Pfaff, F., +, *JAIF*, **11**, 1 (June 2016), 206–226.

N

Naval Anti-Air Warfare

Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.

nearest neighbor

Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.

neural networks

Belaroussi, R., +, *JAIF*, **1**, 1 (July 2006), 35–51.

Nonlinear filtering

Duník, J., +, *JAIF*, **11**, 1 (June 2016), 91–109.

Jilkov, V. P., +, *JAIF*, **10**, 1 (June 2015), 73–00.

Kurz, G., +, *JAIF*, **11**, 1 (June 2016), 138–156.

Nonlinear Kalman Filtering

Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.

non-uniform quantization

Habtemariam, B. K., +, *JAIF*, **7**, 2 (December 2012), 114–130.

Numerical Derivatives

Roth, M., +, *JAIF*, **11**, 1 (June 2016), 47–70.

O

observability

Belfadel, D., +, *JAIF*, **10**, 2 (December 2015), 101–112.

OFDM modulation

Choi, S., +, *JAIF*, **9**, 1 (July 2014), 27–37.

Organizational Structures Identification

Han, X., +, *JAIF*, **7**, 1 (June 2012), 78–96.

OTHR

Romeo, K., +, *JAIF*, **10**, 2 (December 2015), 113–124.

out-of-sequence measurement

Zhang, S., +, *JAIF*, **6**, 1 (June 2011), 3–23.

P

parallel and distributed computing, GPU

Jilkov, V. P., +, *JAIF*, **10**, 1 (June 2015), 73–00.

particle filter

Jilkov, V. P., +, *JAIF*, **10**, 1 (June 2015), 73–00.

particle filtering

Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.

Said, S., +, *JAIF*, **11**, 1 (June 2016), 227–249.

particle flow filter

Jilkov, V. P., +, *JAIF*, **10**, 1 (June 2015), 73–00.

passive acoustic sensor network

Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.

Passive radar

Choi, S., +, *JAIF*, **9**, 1 (July 2014), 27–37.

passive sensor

Zhang, S., +, *JAIF*, **9**, 1 (July 2014), 38–46.

PCRLB, LFM

Song, X., +, *JAIF*, **7**, 1 (June 2012), 16–27.

PDA

Panakkal, V. P., +, *JAIF*, **10**, 2 (December 2015), 143–162.

performance analysis

Schlosser, M. S., +, *JAIF*, **2**, 2 (Dec. 2007), 65–76.

performance evaluation

Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.

performance model

Carvalho, R. N., +, *JAIF*, **7**, 2 (December 2012), 141–152.

Performance Prediction

Blair, W. D., +, *JAIF*, **7**, 1 (June 2012), 28–45.

periodic risks

Nitzan, E., +, *JAIF*, **11**, 1 (June 2016), 123–137.

PMHT

Crouse, D. F., +, *JAIF*, **4**, 2 (Dec. 2009), 93–116.

point cloud

Kurz, G., +, *JAIF*, **9**, 1 (July 2014), 13–26.

point processes

Streit, R., *JAIF*, **8**, 2 (December 2013), 119–132.

Possibilistic Reasoning

Alsun, M., +, *JAIF*, 7, 2 (December 2012), 101–113.

posterior Cramer-Rao lower bound

Habtemariam, B. K., +, *JAIF*, 7, 2 (December 2012), 114–130.

predetection Fusion

Georgescu, R., +, *JAIF*, 7, 1 (June 2012), 61–77.

probabilistic data association

Marković, I., +, *JAIF*, 11, 1 (June 2016), 157–172.

Probabilistic Multi-Hypothesis Tracker (PMHT)

Choi, S., +, *JAIF*, 9, 1 (July 2014), 27–37.

probability

Areta, J., +, *JAIF*, 2, 2 (Dec. 2007), 113–127.

Ferry, J. P., *JAIF*, 5, 1 (July 2010), 41–67.

Probability density estimation

Le Bihan, N., +, *JAIF*, 11, 1 (June 2016), 173–185.

Probability generating function

Streit, R., +, *JAIF*, 9, 1 (July 2014), 3–12.

Probability generating functional

Streit, R., +, *JAIF*, 9, 1 (July 2014), 3–12.

probability hypothesis density

Streit, R., *JAIF*, 8, 2 (December 2013), 119–132.

Probability hypothesis density filter

Habtemariam, B. K., +, *JAIF*, 7, 2 (December 2012), 114–130.

profile-free method

Tharmarasa, R., +, *JAIF*, 7, 1 (June 2012), 46–60.

proportional conflict redistribution rules

Martin, A., +, *JAIF*, 3, 2 (Dec. 2008), 67–89.

Q

Quadratic Assignment Problem

Han, X., +, *JAIF*, 7, 1 (June 2012), 78–96.

qualitative beliefs

Martin, A., +, *JAIF*, 3, 2 (Dec. 2008), 67–89.

quantized

Stone, L. D., +, *JAIF*, 10, 1 (June 2015), 3–12.

R

radar

Kahler, B., +, *JAIF*, 6, 2 (December 2011), 101–118.

random sets

Stone, L. D., +, *JAIF*, 10, 1 (June 2015), 3–12.

Real-time Fusion

Sudit, M., +, *JAIF*, 2, 1 (July 2007), 3–21.

realistic sensor data

Valin, P., +, *JAIF*, 5, 1 (July 2010), 32–40.

reasoning

Biermann, J., +, *JAIF*, 8, 2 (December 2013), 101–118.

recursive Bayesian estimation

Pfaff, F., +, *JAIF*, 11, 1 (June 2016), 206–226.

recursive filtering

Kurz, G., +, *JAIF*, 9, 2 (December 2014), 90–105.

reflection

Song, X., +, *JAIF*, 8, 1 (July 2013), 3–14.

resource allocation

Avasarala, V., +, *JAIF*, 4, 1 (July 2009), 52–71.

resource management

Carvalho, R. N., +, *JAIF*, 7, 2 (December 2012), 141–152.

Johansson, F., +, *JAIF*, 6, 2 (December 2011), 187–199.

restless bandits

Washburn, R. B., +, *JAIF*, 3, 1 (July 2008), 3–13.

risk

Biermann, J., +, *JAIF*, 8, 2 (December 2013), 101–118.

road map

Yang, C., +, *JAIF*, 3, 1 (July 2008), 14–32.

road network extraction

Salerno, E., +, *JAIF*, 10, 1 (June 2015), 58–72.

rollout algorithm

An, W., +, *JAIF*, 3, 1 (July 2008), 33–49.

Rotation group $SO(3)$

Le Bihan, N., +, *JAIF*, 11, 1 (June 2016), 173–185.

S

S^2 KF

Steinbring, J., +, *JAIF*, 9, 2 (December 2014), 106–123.

S-D algorithm

Zhang, S., +, *JAIF*, 9, 1 (July 2014), 38–46.

sample-based stochastic filtering via root-finding (SB-SFRF)

Nitzan, E., +, *JAIF*, 11, 1 (June 2016), 123–137.

Sampling

Steinbring, J., +, *JAIF*, 9, 2 (December 2014), 106–123.

Schmidt-Kalman filter

Zhang, S., +, *JAIF*, 6, 1 (June 2011), 3–23.

score fusion evaluation metric

Zheng, Y., +, *JAIF*, 9, 2 (December 2014), 124–135.

Secular function

Streit, R., +, *JAIF*, 9, 1 (July 2014), 3–12.

Selection Bias

Junghans, M., +, *JAIF*, 3, 1 (July 2008), 50–62.

self-fusion

Howell, C., +, *JAIF*, 8, 2 (December 2013), 133–142.

sensor assignment

An, W., +, *JAIF*, 3, 1 (July 2008), 33–49.

sensor fusion

Blair, W. D., +, *JAIF*, 7, 1 (June 2012), 28–45.

Carvalho, R. N., +, *JAIF*, 7, 2 (December 2012), 141–152.

Sensor Layout

Erdinc, O., +, *JAIF*, 2, 1 (July 2007), 22–34.

sensor localization

Crouse, D. F., +, *JAIF*, 8, 1 (July 2013), 73–89.

sensor management

Avasarala, V., +, *JAIF*, 4, 1 (July 2009), 52–71.

Washburn, R. B., +, *JAIF*, 3, 1 (July 2008), 3–13.

Sensor network

Song, X., +, *JAIF*, 8, 1 (July 2013), 3–14.

Sensor Networks

Georgescu, R., +, *JAIF*, 7, 1 (June 2012), 61–77.

sensor resource management

Chang, K. C., +, *JAIF*, 1, 2 (Dec. 2006), 95–107.

sensor scheduling

An, W., +, *JAIF*, 3, 1 (July 2008), 33–49.

Shooter localization

George, J., +, *JAIF*, 8, 1 (July 2013), 15–32.

Sigma-Points

Roth, M., +, *JAIF*, 11, 1 (June 2016), 47–70.

Similarity Estimation

Alsun, M., +, *JAIF*, 7, 2 (December 2012), 101–113.

similarity index

Jabeur, K., +, *JAIF*, 4, 2 (Dec. 2009), 75–92.

Simulation

Crouse, D. F., +, *JAIF*, 10, 1 (June 2015), 31–57.

situation assessment

Chen, G., +, *JAIF*, 2, 1 (July 2007), 35–48.

Situation Awareness

Blasch, E., +, *JAIF*, 1, 2 (Dec. 2006), 122–143.

Brynielsson, J., +, *JAIF*, 1, 2 (Dec. 2006), 108–121.

situational assessment

Sudit, M., +, *JAIF*, 2, 1 (July 2007), 3–21.

situational awareness

Sudit, M., +, *JAIF*, 2, 1 (July 2007), 3–21.

smoothing

Chakravorty, R., +, *JAIF*, 1, 1 (July 2006), 63–74.

social networks

Biermann, J., +, *JAIF*, 8, 2 (December 2013), 101–118.

sonar

Erdinc, O., +, *JAIF*, 2, 1 (July 2007), 22–34.

space-time adaptive processing (STAP)

Kahler, B., +, *JAIF*, 6, 2 (December 2011), 101–118.

space tracking

Belfadel, D., +, *JAIF*, 10, 2 (December 2015), 101–112.

spline

Kurz, G., +, *JAIF*, 9, 1 (July 2014), 13–26.

Stansfield estimator

Song, X., +, *JAIF*, 8, 1 (July 2013), 3–14.

state constraints

Yang, C., +, *JAIF*, 3, 1 (July 2008), 14–32.

state estimation

Duník, J., +, *JAIF*, 11, 1 (June 2016), 91–109.

statistical efficiency

Belfadel, D., +, *JAIF*, 9, 2 (December 2014), 59–74.

Belfadel, D., +, *JAIF*, 10, 2 (December 2015), 101–112.

Osborne, III, R. W., +, *JAIF*, 10, 2 (December 2015), 199–210.

stereo camera

Kurz, G., +, *JAIF*, 9, 1 (July 2014), 13–26.

stochastic filtering

Said, S., +, *JAIF*, **11**, 1 (June 2016), 227–249.

stochastic systems

Duník, J., +, *JAIF*, **11**, 1 (June 2016), 91–109.

structure discovery

Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.

structured hypotheses

Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.

subgraph matching

Sambhoos, K., +, *JAIF*, **3**, 2 (Dec. 2008), 90–106.

submodular minimization

Papageorgiou, D. J., +, *JAIF*, **6**, 2 (December 2011), 77–100.

sudden maneuvers

Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.

surface estimation

Kurz, G., +, *JAIF*, **9**, 1 (July 2014), 13–26.

surveillance

Tian, X., +, *JAIF*, **5**, 1 (July 2010), 3–17.

T**target attributes**

Bar-Shalom, Y., +, *JAIF*, **2**, 1 (July 2007), 49–59.

Target Engageability

Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.

Target Existence Uncertainty

Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.

target tracking

Aughenbaugh, J. M., +, *JAIF*, **5**, 2 (Dec. 2010), 108–127.

Bar-Shalom, Y., +, *JAIF*, **2**, 1 (July 2007), 49–59.

Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.

Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.

Crouse, D. F., +, *JAIF*, **10**, 1 (June 2015), 31–57.

George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.

Jilkov, V. P., +, *JAIF*, **10**, 1 (June 2015), 73–80.

Zhang, S., +, *JAIF*, **6**, 1 (June 2011), 3–23.

target-death problem

Coraluppi, S., +, *JAIF*, **7**, 2 (December 2012), 153–164.

TBM, Monte Carlo simulation

Hamp, Q., +, *JAIF*, **7**, 2 (December 2012), 131–140.

templates

Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.

test sequencing

Choi, S., +, *JAIF*, **8**, 2 (December 2013), 143–155.

textual representation

Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.

threat evaluation

Johansson, F., +, *JAIF*, **6**, 2 (December 2011), 187–199.

threat prediction

Chen, G., +, *JAIF*, **2**, 1 (July 2007), 35–48.

total least squares

Salerno, E., +, *JAIF*, **10**, 1 (June 2015), 58–72.

track association

Bar-Shalom, Y., +, *JAIF*, **3**, 2 (Dec. 2008), 107–117.

Osborne, R. W., III, +, *JAIF*, **7**, 1 (June 2012), 3–15.

track detection

Romeo, K., +, *JAIF*, **10**, 2 (December 2015), 113–124.

track fusion

Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.

Bar-Shalom, Y., +, *JAIF*, **3**, 2 (Dec. 2008), 107–117.

Yang, C., +, *JAIF*, **3**, 1 (July 2008), 14–32.

track initialization

Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.

track-before-detect

Davey, S. J., *JAIF*, **6**, 2 (December 2011), 167–186.

Track-to-Track

Tian, X., +, *JAIF*, **4**, 2 (Dec. 2009), 146–164.

Track-to-track association

Papageorgiou, D. J., +, *JAIF*, **6**, 2 (December 2011), 77–100.

tracking

An, W., +, *JAIF*, **3**, 1 (July 2008), 33–49.

Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.

Aughenbaugh, J. M., +, *JAIF*, **5**, 2 (Dec. 2010), 108–127.

Bar-Shalom, Y., +, *JAIF*, **1**, 1 (July 2006), 3–14.

Bar-Shalom, Y., +, *JAIF*, **2**, 1 (July 2007), 49–59.

Bar-Shalom, Y., +, *JAIF*, **3**, 2 (Dec. 2008), 107–117.

Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.

Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.

Crouse, D. F., +, *JAIF*, **4**, 2 (Dec. 2009), 93–116.

Erdinc, O., +, *JAIF*, **2**, 1 (July 2007), 22–34.

Jilkov, V. P., +, *JAIF*, **4**, 1 (July 2009), 27–39.

Kurz, G., +, *JAIF*, **9**, 1 (July 2014), 13–26.

Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.

Rodningsby, A., +, *JAIF*, **4**, 2 (Dec. 2009), 117–145.

Song, X., +, *JAIF*, **7**, 1 (June 2012), 16–27.

Tian, X., +, *JAIF*, **4**, 2 (Dec. 2009), 146–164.

Tian, X., +, *JAIF*, **5**, 2 (Dec. 2010), 128–138.

tracking and classification

Carvalho, R. N., +, *JAIF*, **7**, 2 (December 2012), 141–152.

tracking, optimisation

Duník, J., +, *JAIF*, **11**, 1 (June 2016), 91–109.

Traffic Surveillance

Junghans, M., +, *JAIF*, **3**, 1 (July 2008), 50–62.

trafficability

George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.

Transferable Belief Model

Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.

U**UAV search**

Jilkov, V. P., +, *JAIF*, **4**, 1 (July 2009), 27–39.

UAV

Tian, X., +, *JAIF*, **5**, 1 (July 2010), 3–17.

uncertainty

Hamp, Q., +, *JAIF*, **7**, 2 (December 2012), 131–140.

Uncertainty Quantification

Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.

unscented transform

Osborne, R. W., III, +, *JAIF*, **7**, 1 (June 2012), 3–15.

Unscented Transformation

Roth, M., +, *JAIF*, **11**, 1 (June 2016), 47–70.

V**visual representation**

Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.

VLO

Romeo, K., +, *JAIF*, **10**, 2 (December 2015), 113–124.

volcanic eruptions

Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.

von Mises-Fisher distribution

Marković, I., +, *JAIF*, **11**, 1 (June 2016), 157–172.

voting algorithms

Hoseinnezhad, R., +, *JAIF*, **1**, 1 (July 2006), 52–62.

W**wake**

Rodningsby, A., +, *JAIF*, **4**, 2 (Dec. 2009), 117–145.

weapon allocation

Johansson, F., +, *JAIF*, **6**, 2 (December 2011), 187–199.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

ISIF Website: <http://www.isif.org>

2017 BOARD OF DIRECTORS*

2013–2015	2014–2016	2015–2017
Jean Dezert	Sten F. Andler	Joachim Biermann
Gee-Wah Ng	Murat Efe	Frederik Gustafsson
Anne-Laure Jousset	Lyudmila Mihaylova	Jesús García

*Board of Directors are elected by the members of ISIF for a three year term.

PAST PRESIDENTS

Jean Dezert, 2016	Stefano Coraluppi, 2010	Xiao-Rong Li, 2003
Darin Dunham, 2015	Elisa Shahbazian, 2009	Yaakov Bar-Shalom, 2002
Darin Dunham, 2014	Darko Musicki, 2008	Pramod Varshney, 2001
Wolfgang Koch, 2013	Erik Blasch, 2007	Yaakov Bar-Shalom, 2000
Roy Streit, 2012	Pierre Valin, 2006	Jim Llinas, 1999
Joachim Biermann, 2011	W. Dale Blair, 2005	Jim Llinas, 1998
	Chee Chong, 2004	

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.