December 2020 Volume 15 Number 2 ISSN 1557-6418

JAIF20

Journal of Advances in Information Fusion

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

Regular Papers Page

From the Associate Editor-in-Chief

The Ubiquity of Multisensor Fusion

www.isif.org

ISIF

INTERNATIONAL SOCIETY OF INFORMATION FUSION

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

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

From the Associate Editor-in-Chief

December 2020

David Frederic Crouse

The Ubiquity of Multisensor Fusion

The topic of multiple sensor fusion is becoming increasingly mainstream. For example, while cellphones have long contained cameras, magnetometers, and accelerometers, they now contain LiDAR (Light Detection and Ranging) [1] as well. A smartphone can even be strapped to one's arm to serve as a goniometer to measure the range of motion of a joint [2]. Even some smartwatches have started sporting small W-band radars for gesture recognition [3].

The cost of sensors is decreasing as their ubiquity is increasing. This means that opportunities for hobbyists abound. For example, while until a few years ago simultaneous localization and mapping (SLAM) was more within the purview of the military, Google has now started open sourcing software for LiDAR-based SLAM [4].

Given the diversity of low-cost sensors available nowadays, it is natural to want to fuse all of the available information. A prerequisite to sensor fusion is that one establish a common coordinate system across sensors. Two papers in this issue address this topic. The paper "Bias Estimation for Collocated Sensors: Model Identification and Measurement Fusion" considers measurement-level registration and fusion of collocated sensors. These could be for autonomous driving, or perhaps for some augmented reality feature in a smartphone. Additionally, the topic of multisensor fusion is taken up with the paper "Statistically Efficient Multisensor Rotational Bias Estimation for Passive Sensors Without Target State Estimation," which addresses fusion between noncollocated passive angle-only sensors.

The third paper of this issue addresses additional challenges of multisensor fusion while handling a variety of target states. Different sensors might not make measurements or produce estimates at the same rate. Additionally, some sensors might produce estimates in different coordinate systems from others. Many off-the-shelf sensors contain built-in trackers, precluding the possibility of a fully centralized tracking architecture. The paper "Heterogeneous and Asynchronous Information Matrix Fusion" addresses such issues. It considers the

asynchronous fusion of tracks from sensors in different coordinate systems.

While the fusion literature primarily focuses on the fusion of single-model estimates across sensors, the fourth paper of this issue, entitled "Track-to-Track Fusion Using Inside Information From Local IMM Estimators," discusses how to fuse estimates from sensors running interactive multiple model (IMM) estimators. Given the mature state of the multiple-model-estimation literature, it is good to see that the time has arrived for an increasing number of multiple-model-fusion algorithms.

Finally, the *ISIF Journal of Advances in Information Fusion* (JAIF) always encourages the submission of expanded papers from the International Conference on Information Fusion. This year's Fusion conference was virtual for the first time due to the COVID-19 pandemic. Hopefully, the worldwide approval and dissemination of COVID-19 vaccines (and who knows, maybe even sniffer dogs [5]) will prove effective in stemming the outbreak and will allow for the 2021 Fusion conference to be held as planned in Sun City, South Africa. I look forward to seeing many members of the fusion community there.

REFERENCES

[1] T. B. Lee
"Lidar used to cost $75,000—here's how Apple brought it to the iPhone," Ars Technica, 15 Oct. 2020. [Online]. Available: https://arstechnica.com/cars/2020/10/the-technology-behind-the-iphone-lidar-may-be-coming-soon-to-cars/.

[2] C. Roldán-Jiménez, J. Martin-Martin, and A. I. Cuesta-Vargas
"Reliability of a smartphone compared with an intertial sensor to measure shoulder mobility: Cross-sectional study," *JMIR mHealth uHealth*, vol. 7, no. 9, Sep. 2019, Art. no. e13640.

[3] J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and P. Ivan
"Soli: Ubiquitous gesture sensing with a millimeter wave radar,"
ACM Trans. Graph., vol. 35, no. 4, Jul. 2016, Art. no. 142.

[4] Google. Cartographer. [Online]. Available: https://opensource.google/projects/cartographer.

[5] H. Else
"Can dogs smell COVID? Here's what the science says," *Nature*, vol. 587, no. 7835, pp. 530–531, Nov. 2020.

Statistically Efficient Multisensor Rotational Bias Estimation for Passive Sensors without Target State Estimation

MICHAEL KOWALSKI
YAAKOV BAR-SHALOM
PETER WILLETT
BENNY MILGROM
RONEN BEN-DOV

In target tracking applications, it is necessary to account for measurement biases present within the sensors. For passive sensors, these biases are commonly represented as unknown rotations of the sensor measurements and must be estimated. As targets may move in unpredictable ways, it is advantageous to decouple target state and sensor bias estimation to simplify the estimation problem. To do this, a bias pseudo-measurement method must be used in which the measurements are converted and differenced to eliminate the presence of the true target state. For passive angle-only sensors, it is important to appropriately convert lines of sight into Cartesian space. By using the closest point of approach method, it is possible to apply the bias pseudo-measurement method to these sensors. The Cramér–Rao lower bound can be obtained for this method, and, furthermore, it can be attained by using a maximum likelihood estimation method.

Manuscript received May 12, 2020; revised May 18, 2020; released for publication July 31, 2020.

Associate Editor: Florian Meyer.
The authors are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA (E-mail: michael.p.kowalski@uconn.edu, ybs@uconn.edu, peter.willett@uconn.edu, milgromb@gmail.com, bdronen@gmail.com).
M. Kowalski and P. Willett were supported by AFOSR under contract FA9500-18-1-0463.

I. INTRODUCTION

In target tracking, it is common that the sensors employed are subject to systematic errors known as sensor measurement biases. Errors present in sensors, such as calibration, alignment, or clock time [3], [12], can contribute to such biases. These errors can also be related to environmental effects such as temperature-due warping of the sensor material and atmospheric refraction. Furthermore, many of the advanced methods developed for target tracking do not take into account these sources of error, which can result in diminished performance. Therefore, it is necessary to use methods to estimate these biases and then remove the corresponding error from the sensor measurements before implementing target tracking solutions.

In the past, there have been primarily two methods of bias estimation that have been implemented. The first is simultaneous target state and sensor bias estimation [4], [6], [7]. In this method, the state of the target is estimated jointly and simultaneously with the sensor biases. A significant problem with this method is that the target as a practical matter needs to be assumed to be moving in a deterministic manner. If not, the target state at all times must be estimated, which is computationally infeasible and prone to numerical problems in ill-conditioned systems [20]. An additional issue is the increase in the dimension of the parameter vector that must be estimated: not just bias parameters but also those of the targets. This is both a computational concern (increased complexity) and one of performance, since more parameters always mean more error. On the other hand, the advantage with this method is that should the state information be known, then it is possible to achieve efficient results using all of the measurement information. Additionally, there are fewer issues with measurement synchronization [21].

The second method for bias estimation is to use bias pseudo-measurements [9], [16], [20]–[22], [26]. In this method, the original measurements are converted into Cartesian within common coordinates [such as Earth-centered Earth-fixed (ECEF)] and then differenced to eliminate the true target state. This process leaves solely the effect of the biases and noise, which are used as the measurements for bias estimation. This can be advantageous because it enables the system to decouple the process of state estimation and bias estimation. The problem with this method is that the removal of the target state information can potentially reduce the effectiveness of bias estimation, as some measurement information can be lost. Additionally, conversions can be nonlinear and result in additional error as the noise is converted.

The sensor biases can be modeled in many forms that depend on the sources of error that affect the types of sensors in question. For example, there are results investigating bias estimation to additive and multiplicative biases [16], [22], [26]. These biases affect the measurements directly by adding an unknown value or multiply-

ing them by an unknown value. In the present work, the biases of angle-only (passive) sensors are explored. As these sensors are line-of-sight (LOS) sensors, the biases present are chosen to be modeled as a rotation of the LOS around the sensor. The sensors provide two angle measurements and have 3-D alignment error. The rotation is a nonlinear Euler rotation using yaw (azimuth), pitch (elevation), and roll (rotation of the field of view), which is a challenge to estimate. Rotational bias estimation has been examined using simultaneous target state and bias estimation in the past, but little has been done for the pseudo-measurement method [9], [19], [26]. In particular, these methods achieve this by using a conversion via LOS triangulation; however, this method has drawbacks as a result of its nonlinear conversion that relies on projecting the LOSs into a single plane [17]. The present work seeks to improve upon the method of [19] by using the closest point of approach (CPA) method of conversion [18], which avoids observability problems by working in 3-D instead of 2-D.

The CPA conversion used here is based on the method of finding the closest point between nonintersecting lines [1], [15]. This is made via a least-squares framework, where the squared distance of a point between two lines is minimized. A cost function is made and differentiated in order to find this point, where the derivative is zero. In three dimensions, this results in simple expression that can be itself differentiated to obtain the Jacobians that are necessary for the bias pseudo-measurement method. This method of conversion differs from maximum likelihood (ML) conversion from LOS to Cartesian as there is no iteration involved, such as in [24], and instead an explicit expression is used.

In previous research, simultaneous target state and bias estimation has often been used to overcome the challenge of a nonlinear bias [4], [6], [7]. However, this method relies upon having a target of opportunity that moves deterministically, and in many applications, it is impossible to predict a target's motion as it may move in nonlinear and maneuvering ways that do not fit the expected target motion. Therefore, it is desirable to decouple the target state and the estimation of sensor bias. This bias pseudo-measurement method has been applied to additive and multiplicative biases in active sensors successfully in previous research [16], [22], [26]. Most work in angle-only LOS sensor bias estimation has been done solely in 2-D bearings-only problems. These methods are limited to 2-D Cartesian space with angle-only sensors and bias only in the one angle. Methods for bistatic measurements have been introduced in [27] and [28]. In [25], the pseudo-measurement method is applied with time-of-arrival measurements to improve the accuracy. In [8], a particle filter is shown to be usable for bias estimation for bearings-only sensors. In [26], it was shown that it is possible to find the Cramér–Rao lower bound (CRLB) but that achieving it is difficult. There has been very little work to apply pseudo-measurement techniques to 3-D passive sensors [9]. The main contribution of the present work over [9] and [26] is to show attainability of the CRLB; i.e., our algorithm is statistically efficient. In addition, [9] is limited to biases in azimuth and elevation, lacking a roll bias.

Once the pseudo-measurements are generated, it is possible to use them to estimate the sensor biases separately from the target state. To estimate the biases in sensors, it is simple and effective to use the ML approach implemented via the iterated least-squares (ILS) method if the biases are constant over a batch of measurements. ILS estimation has been used in 3-D passive sensor [5] and 3-D spherical sensor bias registration. In this paper, ILS is used to estimate the rotational biases in 3-D passive sensors.

The outline of this paper is as follows. The passive sensor model is defined in Section II. In Section III, the passive sensor angle-only measurements are converted into Cartesian coordinates. In Section IV, the pseudo-measurement approach for estimating the biases is outlined in Section IV-A and the ML estimation described in Section IV-B. The CRLB is presented in Section V. Simulation results presented in Section VI show that the proposed method yields sensor bias estimates that meet the CRLB. Section VII concludes the paper.

II. PROBLEM FORMULATION

The problem formulation for this work involves target tracking using passive angle-only sensors in 3-D Cartesian space. There are N_t targets of opportunity and N_s sensors that move over K time steps. The common Cartesian reference frame is in ECEF coordinates. The position of each sensor s, which is assumed to be known by the network, is defined as

$$\boldsymbol{x}_s(k) = [x_s(k),\ y_s(k),\ z_s(k)]^{\mathrm{T}}. \tag{1}$$

These sensors are assumed synchronous. Each target t has a position in the common Cartesian frame unknown to the network, given by

$$\boldsymbol{x}^t(k) = [x^t(k),\ y^t(k),\ z^t(k)]^{\mathrm{T}}. \tag{2}$$

The targets can move in arbitrary ways, but their positions related to the measurement origin must be known for all times. Each sensor has an LOS to the targets based in its own reference frame. The position of the target t with respect to the sensor s in the common Cartesian frame translates to the sensor location as

$$\boldsymbol{x}_s^t(k) = \boldsymbol{x}^t(k) - \boldsymbol{x}_s(k). \tag{3}$$

The sensor reference frame is rotated (with respect to the common Cartesian frame) using the Euler angle rotation method. The sensors are affected by the known nominal rotation $\boldsymbol{\omega}_s^n$ and the unknown bias rotation $\boldsymbol{\omega}_s^b$. The target position in the rotated sensor frame is then

$$\boldsymbol{x}_s^{t,n,b}(k) = T_s(\boldsymbol{\omega}_s^n)T_s(\boldsymbol{\omega}_s^b)\left(\boldsymbol{x}^t(k) - \boldsymbol{x}_s(k)\right). \tag{4}$$

The biases consist of yaw, pitch, and roll, defined as θ, ϕ, and ψ, respectively. For clarity, the superscripts n and

b are used to denote rotation variables for the nominal rotation and bias rotation, respectively. A variable with both superscripts is rotated by both. The rotations for sensor s are defined as

$$\boldsymbol{\omega}_s^n = \begin{bmatrix} \theta_s^n & \phi_s^n & \psi_s^n \end{bmatrix}^{\mathrm{T}}, \tag{5}$$

$$\boldsymbol{\omega}_s^b = \begin{bmatrix} \theta_s^b & \phi_s^b & \psi_s^b \end{bmatrix}^{\mathrm{T}}, \tag{6}$$

$$T(\boldsymbol{\omega}_s^i) = T(\theta_s^i, \phi_s^i, \psi_s^i) = T_\theta(\theta_s^i) T_\phi(\phi_s^i) T_\psi(\psi_s^i) =$$

$$\begin{bmatrix} \cos(\theta)\cos(\phi) & \cos(\theta)\sin(\phi)\sin(\psi) - \sin(\theta)\cos(\psi) & \cos(\theta)\sin(\phi)\cos(\psi) + \sin(\theta)\sin(\psi) \\ \sin(\theta)\cos(\phi) & \sin(\theta)\sin(\phi)\sin(\psi) + \cos(\theta)\cos(\psi) & \sin(\theta)\sin(\phi)\cos(\psi) - \cos(\theta)\sin(\psi) \\ -\sin(\phi) & \cos(\phi)\sin(\psi) & \cos(\phi)\cos(\psi) \end{bmatrix},$$

$$i = n, b. \tag{7}$$

The rotated positions (4) that are used by the sensors produce the rotated azimuth and elevation measurements (represented by ξ for the vector of azimuth α and elevation ϵ, respectively)

$$\begin{aligned} \xi_s^{t,n,b}(k) &= \begin{bmatrix} \alpha_s^{t,n,b}(k) \\ \epsilon_s^{t,n,b}(k) \end{bmatrix} \\ &= \begin{bmatrix} \tan^{-1}\left(\frac{y_s^{t,n,b}(k)}{x_s^{t,n,b}(k)}\right) \\ \tan^{-1}\left(\frac{z_s^{t,n,b}(k)}{\sqrt{x_s^{t,n,b}(k)^2 + y_s^{t,n,b}(k)^2}}\right) \end{bmatrix}. \end{aligned} \tag{8}$$

Uncorrelated (across sensors), independent (across time), zero-mean, white Gaussian noise is added to obtain the measurements, denoted by $w_s^{t,n,b,\alpha}(k)$ and $w_s^{t,n,b,\epsilon}(k)$ for azimuth and elevation, respectively. These noises have variances $(\sigma_s^\alpha)^2$ and $(\sigma_s^\epsilon)^2$. An expansion is used to approximate the effect of the nominal rotation in equation (5) and biases in equation (6) through the use of Jacobians (see Appendix A). This results in biased and noisy measurements, denoted by ζ, with the measurement equation

$$\begin{aligned} \zeta_s^{t,n,b}(k) &= \begin{bmatrix} \alpha_s^{t,n,b}(k) \\ \epsilon_s^{t,n,b}(k) \end{bmatrix} + \begin{bmatrix} w_s^{t,\alpha}(k) \\ w_s^{t,\epsilon}(k) \end{bmatrix} \\ &\approx \begin{bmatrix} \alpha_s^{t,b}(k) \\ \epsilon_s^{t,b}(k) \end{bmatrix} + \begin{bmatrix} w_s^{t,\alpha}(k) \\ w_s^{t,\epsilon}(k) \end{bmatrix} \\ &\quad + \begin{bmatrix} \frac{\partial \alpha_s^{t,n,b}(k)}{\partial \theta_s^n} & \frac{\partial \alpha_s^{t,n,b}(k)}{\partial \phi_s^n} & \frac{\partial \alpha_s^{t,n,b}(k)}{\partial \psi_s^n} \\ \frac{\partial \epsilon_s^{t,n,b}(k)}{\partial \theta_s^n} & \frac{\partial \epsilon_s^{t,n,b}(k)}{\partial \phi_s^n} & \frac{\partial \epsilon_s^{t,n,b}(k)}{\partial \psi_s^n} \end{bmatrix} \begin{bmatrix} \theta_s^n \\ \phi_s^n \\ \psi_s^n \end{bmatrix} \\ &\approx \begin{bmatrix} \alpha_s^t(k) \\ \epsilon_s^t(k) \end{bmatrix} + \begin{bmatrix} w_s^{t,\alpha}(k) \\ w_s^{t,\epsilon}(k) \end{bmatrix} \\ &\quad + \begin{bmatrix} \frac{\partial \alpha_s^{t,b}(k)}{\partial \theta_s^b} & \frac{\partial \alpha_s^{t,b}(k)}{\partial \phi_s^b} & \frac{\partial \alpha_s^{t,b}(k)}{\partial \psi_s^b} \\ \frac{\partial \epsilon_s^{t,b}(k)}{\partial \theta_s^b} & \frac{\partial \epsilon_s^{t,b}(k)}{\partial \phi_s^b} & \frac{\partial \epsilon_s^{t,b}(k)}{\partial \psi_s^b} \end{bmatrix} \begin{bmatrix} \theta_s^b \\ \phi_s^b \\ \psi_s^b \end{bmatrix} \\ &\quad + \begin{bmatrix} \frac{\partial \alpha_s^{t,n,b}(k)}{\partial \theta_s^n} & \frac{\partial \alpha_s^{t,n,b}(k)}{\partial \phi_s^n} & \frac{\partial \alpha_s^{t,n,b}(k)}{\partial \psi_s^n} \\ \frac{\partial \epsilon_s^{t,n,b}(k)}{\partial \theta_s^n} & \frac{\partial \epsilon_s^{t,n,b}(k)}{\partial \phi_s^n} & \frac{\partial \epsilon_s^{t,n,b}(k)}{\partial \psi_s^n} \end{bmatrix} \begin{bmatrix} \theta_s^n \\ \phi_s^n \\ \psi_s^n \end{bmatrix} \\ &\triangleq \xi_s^t(k) + C_s^{t,b}(k)\boldsymbol{\omega}_s^b + C_s^{t,n,b}(k)\boldsymbol{\omega}_s^n + \boldsymbol{w}_s^t(k), \end{aligned} \tag{9}$$

with the measurement noises for the sensor LOS angles being

$$w_s^{t,\alpha}(k) \sim \mathcal{N}(0, (\sigma_s^\alpha)^2), \qquad w_s^{t,\epsilon}(k) \sim \mathcal{N}(0, (\sigma_s^\epsilon)^2). \tag{10}$$

The matrix $C_s^{t,n,b}$ is the Jacobian of the sensor LOS angles (at the nominal and bias rotation) with respect to the nominal rotation. The matrix $C_s^{t,b}$ is the Jacobian of the sensor LOS angles (at the nominal rotation) with respect to the bias rotation. The corresponding partial derivatives are given in Appendix A. These measurements are assumed to be synchronous, although an asynchronous extension is possible [21]. It is important to note that this Taylor series expansion is an approximation, and in certain cases, the nonlinearity may cause additional error. It is possible to add higher order elements into the expansion, as this is only a first-order expansion, in order to reduce this error.

III. CONVERSION INTO THE COMMON CARTESIAN COORDINATES

In order to produce bias pseudo-measurements, it is necessary to first convert the angle-only measurements into the common frame of reference. This is done by converting them into the common Cartesian coordinates. This can be done using the triangulation method [17] and CPA [18]. In [19], the pseudo-measurement method was originally proposed for passive sensors, albeit with the planar triangulation method. The method was successful; however, it required a relatively low noise standard deviation. For this work, the CPA method is considered as it is more accurate for converting to Cartesian [18], even achieving the CRLB of coordinate conversion. This is much simpler than the ML method used for conversion as in [24], which requires a numerical search and does not yield the necessary Jacobians.

Before the conversion to the common Cartesian coordinates can be made, it is necessary to remove the nominal rotation so that the LOSs are in the same common Cartesian coordinate reference frame. This rotation is known and can be removed by inverting it using the function h^i, defined in equation (13). This calculation is included in Appendix A. It is also necessary to account for this in the noise covariance through the Jacobian D that transforms the sensor LOS angle noises into the

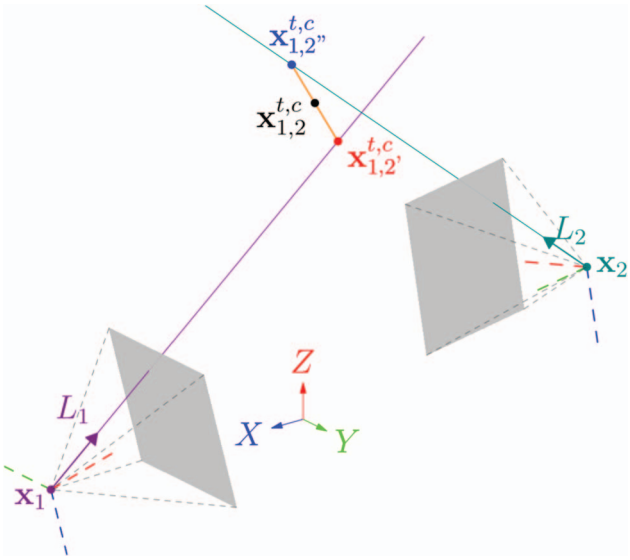


Fig. 1. Using CPA to convert azimuth measurements into 3-D Cartesian measurements. The sensors have their own rotated Cartesian frames with respect to the common Cartesian frame (ECI or ECEF) shown in the center. The LOS measurements are present as rays L_1 and L_2. The two closest positions on the LOSs are found, $\boldsymbol{x}_{1,2'}^{t,c}(k)$ and $\boldsymbol{x}_{1,2''}^{t,c}(k)$, with respect to the common frame and the midpoint $\boldsymbol{x}_{1,2}^{t,c}(k)$ is accepted as the measurement of the target position.

rotated LOS noises (this rotation converts the sensor LOS angles into the common Cartesian system angles). This Jacobian is

$$D_s^{t,n,b}(k) = \begin{bmatrix} \dfrac{\partial \alpha_s^{t,b}(k)}{\partial \alpha_s^{t,n,b}(k)} & \dfrac{\partial \alpha_s^{t,b}(k)}{\partial \epsilon_s^{t,n,b}(k)} \\ \dfrac{\partial \epsilon_s^{t,b}(k)}{\partial \alpha_s^{t,n,b}(k)} & \dfrac{\partial \epsilon_s^{t,b}(k)}{\partial \epsilon_s^{t,n,b}(k)} \end{bmatrix}. \tag{11}$$

The converted noise in the common Cartesian system angles is

$$\boldsymbol{w}_s^{t,b}(k) = D_s^{t,n,b}(k)\boldsymbol{w}_s^t(k). \tag{12}$$

The angle measurement equation in the common Cartesian frame is

$$\zeta_s^{t,b}(k) = h^i\left(\zeta_s^{t,n,b}(k), \boldsymbol{\omega}_s^n\right) \tag{13}$$

$$\approx \xi_s^{t,b}(k) + \boldsymbol{w}_s^{t,b}(k) \tag{14}$$

$$\approx \xi_s^t(k) + C_s^{t,b}(k)\boldsymbol{\omega}_s^b + D_s^{t,n,b}(k)\boldsymbol{w}_s^t(k). \tag{15}$$

The expanded definition of equation (13) and the individual partial derivatives are given in Appendix A, where the approximation is the Taylor series to first order.[1] The CPA method uses two LOSs and finds for each LOS the closest Cartesian positions along the other LOS. The midpoint of these two points is the CPA and can be accepted as a measurement of the Cartesian position of the target. This process is illustrated in Fig. 1. Normally, the midpoint of these positions is used as a single Cartesian measurement; however it is useful for bias estimation to keep these positions separate in order to improve the diversity of the pseudo-measurements. The superscript c is used to indicate conversion via closest point of approach, which is calculated as

$$\boldsymbol{x}_{12'}^{t,c}(k) = \begin{bmatrix} x_{12'}^{t,c}(k) \\ y_{12'}^{t,c}(k) \\ z_{12'}^{t,c}(k) \end{bmatrix} = \boldsymbol{x}_1(k) + \boldsymbol{\lambda}_1^t(k)\frac{(\boldsymbol{\lambda}_1^t(k)'\mathbf{p}_{1,2}(k)) - (\boldsymbol{\lambda}_1^t(k)'\boldsymbol{\lambda}_2^t(k))(\boldsymbol{\lambda}_2^t(k)'\mathbf{p}_{1,2}(k))}{1-(\boldsymbol{\lambda}_1^t(k)'\boldsymbol{\lambda}_2^t(k))^2}, \tag{16}$$

$$\boldsymbol{x}_{12''}^{t,c}(k) = \begin{bmatrix} x_{12''}^{t,c}(k) \\ y_{12''}^{t,c}(k) \\ z_{12''}^{t,c}(k) \end{bmatrix} = \boldsymbol{x}_2(k) + \boldsymbol{\lambda}_2^t(k)\frac{(\boldsymbol{\lambda}_1^t(k)'\boldsymbol{\lambda}_2^t(k))(\boldsymbol{\lambda}_1^t(k)'\mathbf{p}_{1,2}(k)) - (\boldsymbol{\lambda}_2^t(k)'\mathbf{p}_{1,2}(k))}{1-(\boldsymbol{\lambda}_1^t(k)'\boldsymbol{\lambda}_2^t(k))^2}, \tag{17}$$

$$\boldsymbol{\lambda}_1^t(k) = \begin{bmatrix} \cos(\alpha_1^t(k))\cos(\epsilon_1^t(k)) \\ \sin(\alpha_1^t(k))\cos(\epsilon_1^t(k)) \\ \sin(\epsilon_1^t(k)) \end{bmatrix}, \tag{18}$$

$$\boldsymbol{\lambda}_2^t(k) = \begin{bmatrix} \cos(\alpha_2^t(k))\cos(\epsilon_2^t(k)) \\ \sin(\alpha_2^t(k))\cos(\epsilon_2^t(k)) \\ \sin(\epsilon_2^t(k)) \end{bmatrix}, \tag{19}$$

$$\mathbf{p}_{1,2}(k) = \boldsymbol{x}_2(k) - \boldsymbol{x}_1(k) = \begin{bmatrix} x_2(k) - x_1(k) \\ y_2(k) - y_1(k) \\ z_2(k) - z_1(k) \end{bmatrix}. \tag{20}$$

In place of the true azimuth and elevation, the conversion h^c is made using the noisy measurements.

$$\begin{bmatrix} \boldsymbol{x}_{12'}^{t,c}(k) \\ \boldsymbol{x}_{12''}^{t,c}(k) \end{bmatrix} = h^c\left(\xi_{1,2}^t(k)\right), \tag{21}$$

$$\zeta_{1,2}^{t,b,c}(k) = h^c\left(\zeta_{1,2}^{t,b}(k)\right). \tag{22}$$

[1]This conversion is an approximation as the presence of the unknown bias may add some error from the nonlinear conversion from the sensor frame to the common frame. The approximation's error is based on the presence of higher order components, which are negligible relative to the biases themselves.

The new noisy Cartesian measurement equation can be rewritten similarly to equation (9) as

$$\zeta_{1,2}^{t,b,c}(k) \approx \begin{bmatrix} \boldsymbol{x}^t(k) \\ \boldsymbol{x}^t(k) \end{bmatrix} + \begin{bmatrix} \boldsymbol{w}_{12'}^{t,c}(k) \\ \boldsymbol{w}_{12''}^{t,c}(k) \end{bmatrix} + B_{1,2}^t(k) \begin{bmatrix} C_1^{t,b}(k) & \mathbf{0} \\ \mathbf{0} & C_2^{t,b}(k) \end{bmatrix} \begin{bmatrix} \theta_1^b \\ \phi_1^b \\ \psi_1^b \\ \theta_2^b \\ \phi_2^b \\ \psi_2^b \end{bmatrix}$$

$$\approx \boldsymbol{x}^{t,E}(k) + B_{1,2}^t(k) C_{1,2}^{t,b}(k) \boldsymbol{\omega}_{1,2}^b + \boldsymbol{w}_{1,2}^{t,b,c}(k). \quad (23)$$

As it is a Taylor series expansion, this equation is an approximation. Depending on the case, higher order expansion via additional Jacobian terms may be necessary to avoid error. The matrix $B_{1,2}^t$ is the Jacobian of the common Cartesian measurements with respect to the LOS angles in the common Cartesian space, which is

$$B_{1,2}^t(k) = \begin{bmatrix} \nabla_{\xi_1^t,\xi_2^t} \boldsymbol{x}_{12'}^{t,c}(k) \\ \nabla_{\xi_1^t,\xi_2^t} \boldsymbol{x}_{12''}^{t,c}(k) \end{bmatrix}. \quad (24)$$

The Jacobian additionally affects the noise, which is

$$\boldsymbol{w}_{1,2}^{t,b,c}(k) = \begin{bmatrix} \boldsymbol{w}_{12'}^{t,c}(k) \\ \boldsymbol{w}_{12''}^{t,c}(k) \end{bmatrix} \approx B_{1,2}^t(k) \begin{bmatrix} \boldsymbol{w}_1^{t,b}(k) \\ \boldsymbol{w}_2^{t,b}(k) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}_{10\times 1}, R_{1,2}^{t,b,c}(k)\right), \quad (25)$$

$$R_{1,2}^{t,b,c}(k) = B_{1,2}^t(k) D_{1,2}^{t,n,b}(k) R_{1,2}^{t,b,n} D_{1,2}^{t,n,b}(k)' B_{1,2}^t(k)', \quad (26)$$

$$R_{1,2}^{t,b,n} = \begin{bmatrix} (\sigma_1^\alpha)^2 & 0 & 0 & 0 \\ 0 & (\sigma_1^\epsilon)^2 & 0 & 0 \\ 0 & 0 & (\sigma_2^\alpha)^2 & 0 \\ 0 & 0 & 0 & (\sigma_2^\epsilon)^2 \end{bmatrix},$$

where

$$D_{1,2}^{t,n,b}(k) = \begin{bmatrix} D_1^{t,n,b}(k) & \mathbf{0} \\ 0 & D_2^{t,n,b}(k) \end{bmatrix}. \quad (27)$$

It is not necessary to calculate the Cartesian target states in order to generate the Jacobian matrices B, C, and D—they are evaluated at the measured angles. The individual derivatives and gradients are given in Appendix A. A higher order conversion may be used similarly to [23] in order to avoid conversion error in the noise covariance matrix as the noise is an approximation via a Taylor series expansion.

IV. BIAS ESTIMATION

A. Generation of the Bias Pseudo-Measurements

The key step of our method is to difference Cartesian measurements from two pairs of sensors in order eliminate the true target state and be left with solely the effect of the biases and noise converted into Cartesian space. As the true Cartesian state is unknown, it is advantageous to remove it from our measurements. This way any error in the estimation of the Cartesian state does not affect the estimation of the biases. The process of converting all of the sensor measurements into a common Cartesian frame allows its removal by simply differencing the measurements. This isolates the error from biases and noise. With the isolated error, it is possible to estimate the biases by attempting to fit the errors to what is expected in terms of the models used for noise and bias. Denoted by superscript p, these "pseudo-measurements" are calculated as

$$\zeta_{1,2,3,4}^{t,p}(k) = \zeta_{1,2}^{t,b,c}(k) - \zeta_{3,4}^{t,b,c}(k) \approx B_{1,2}^t(k) C_{1,2}^{t,b}(k) \boldsymbol{\omega}_{1,2}^b - B_{3,4}^t(k) C_{3,4}^{t,b}(k) \boldsymbol{\omega}_{3,4}^b + \boldsymbol{w}_{1,2}^{t,b,c}(k) - \boldsymbol{w}_{3,4}^{t,b,c}(k). \quad (28)$$

This can be restructured into a new measurement equation similar to equations (10) and (23), where

$$\zeta_{1,2,3,4}^{t,p}(k) \approx H_{1,2,3,4}^{t,p}(k) \begin{bmatrix} \boldsymbol{\omega}_{1,2}^b \\ \boldsymbol{\omega}_{3,4}^b \end{bmatrix} + \boldsymbol{w}_{1,2,3,4}^{t,p}(k), \quad (29)$$

$$\boldsymbol{w}_{1,2,3,4}^{t,p}(k) \approx \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, R_{1,2,3,4}^{t,p}(k) \right), \quad (30)$$

$$H_{1,2,3,4}^{t,p}(k) = \begin{bmatrix} B_{1,2}^t(k) C_{1,2}^{t,b}(k) & -B_{3,4}^t(k) C_{3,4}^{t,b}(k) \end{bmatrix}, \quad (31)$$

$$R_{1,2,3,4}^{t,p}(k) = R_{1,2}^{t,b,c}(k) + R_{3,4}^{t,b,c}(k). \quad (32)$$

The subscript for parameters (1,2,3,4) denotes that the parameter includes information from the four sensors. The pseudo-measurements are considered an approximation as a result of the previous Taylor series expansions.

B. Maximum Likelihood Estimation of the Biases

It is possible to estimate the biases by using the bias pseudo-measurements, and there are various methods for this. In this paper, we seek the ML estimate (MLE) for the biases, and note that it is desirable to accompany the MLE with the CRLB, since when the MLE is efficient (we will check this) its error performance tracks the CRLB closely. To achieve this, first the measurements

are stacked into a batch

$$\boldsymbol{\zeta} = \left[\zeta^{1,p}_{1,2,3,4}(1), \ldots, \zeta^{N_t,p}_{1,2,3,4}(1), \zeta^{1,p}_{1,2,3,4}(2), \ldots, \zeta^{N_t,p}_{1,2,3,4}(K)\right]^T. \tag{33}$$

The Jacobians and noise covariances are also stacked into matrices $\boldsymbol{H}$ and $\boldsymbol{R}$, respectively, as

$$\boldsymbol{H}^j = \left[H^{1,p}_{1,2,3,4}(1)^j, \ldots, H^{N_t,p}_{1,2,3,4}(1)^j, \ldots, H^{N_t,p}_{1,2,3,4}(K)^j\right]^T, \tag{34}$$

$$\boldsymbol{R}^j = \begin{bmatrix} R^{1,p}_{1,2,3,4}(1)^j & \ldots & 0 \\ \ldots & \ldots & \ldots \\ 0 & \ldots & R^{N_t,p}_{1,2,3,4}(K)^j \end{bmatrix}. \tag{35}$$

As the Jacobian $\boldsymbol{H}$ is calculated using the biased LOS measurements, it is necessary to recalculate it using the debiased measurements as the biases are estimated. This means an iterative method is required—the ILS implementation of the MLE is used. In this method, an initial estimate of zero bias is used and is iteratively updated until the bias estimate converges. Denoting the current ILS iteration by the superscript j,

$$\boldsymbol{\omega}^b_{1,2,3,4} = \left[(\boldsymbol{\omega}^b_1)^T(\boldsymbol{\omega}^b_2)^T(\boldsymbol{\omega}^b_3)^T(\boldsymbol{\omega}^b_4)^T\right] \tag{36}$$

$$\hat{\boldsymbol{\omega}}^{b,(j+1)}_{1,2,3,4} = \hat{\boldsymbol{\omega}}^{b,j}_{1,2,3,4} + [(\boldsymbol{H}^j)'(\boldsymbol{R}^j)^{-1}(\boldsymbol{H}^j)]^{-1}(\boldsymbol{H}^j)'(\boldsymbol{R}^j)^{-1}[\boldsymbol{\zeta} - \boldsymbol{H}^j\hat{\boldsymbol{\omega}}^{b,j}_{1,2,3,4}], \tag{37}$$

$$\hat{\boldsymbol{\omega}}^{b,j=0}_{1,2,3,4} = [0, 0, \ldots, 0]^T. \tag{38}$$

V. CRAMÉR–RAO LOWER BOUND

In order to understand the performance of this bias estimation method, it is necessary to derive a metric for accuracy. The CRLB offers a lower bound on the covariance of an unbiased estimator of a fixed parameter, and hence the root-mean-square error (RMSE) of our method can be compared to it to test for statistical efficiency. Additionally, the performances of other estimation methods can be compared to the present method using this metric. For example, a simultaneous target state and bias estimation method can be compared to this method, which removes the need to estimate the target state. The CRLB is calculated by taking the inverse of the Fisher information matrix

$$J = \boldsymbol{H}'\boldsymbol{R}^{-1}\boldsymbol{H}, \tag{39}$$

that is,

$$\text{CRLB} = J^{-1} = (\boldsymbol{H}'\boldsymbol{R}^{-1}\boldsymbol{H})^{-1}. \tag{40}$$

To find the variances for the individual bias estimates, it is necessary to examine the diagonal elements of the CRLB, $(\sigma_i^{\text{CRLB}})^2$. In the case of approximations—such as those we use here—it may be that an efficient result is not obtained. Otherwise, the bound is, in theory, attained asymptotically. In the case of this work, the CRLB covariance is accepted via hypothesis testing at 5% error. For the estimator to be efficient, the RMSE must be equal to σ_i^{CRLB}. To evaluate the estimator rigorously, its RMSE σ_i for each component is compared to the 95% probability interval of the square root of the CRLB calculated as

$$P(a < \sigma_i < b) = 0.95, \tag{41}$$

$$a = \sigma_i^{\text{CRLB}} - 1.96 \cdot \frac{\sigma_i}{\sqrt{n_{\text{MC}}}}, \tag{42}$$

$$b = \sigma_i^{\text{CRLB}} + 1.96 \cdot \frac{\sigma_i}{\sqrt{n_{\text{MC}}}}, \tag{43}$$

where σ_i is the standard deviation of error in component i from n_{MC} Monte Carlo runs.

The normalized estimator error squared (NEES) [2] can also be evaluated with the chi-square test to verify consistency. The NEES for each Monte Carlo Run is

$$\varepsilon^i = (\boldsymbol{\omega}^b_{1,2,3,4} - \hat{\boldsymbol{\omega}}^{b,i}_{1,2,3,4})'\boldsymbol{J}^i(\boldsymbol{\omega}^b_{1,2,3,4} - \hat{\boldsymbol{\omega}}^{b,i}_{1,2,3,4}), \quad i = 1, 2, \ldots, n_{\text{MC}}. \tag{44}$$

The estimator is considered efficient if the mean of the NEES (multiplied by the number of Monte Carlo runs) lies within the 95% probability region for a chi-square variable with degrees of freedom equal to the number of bias variables multiplied by the number of Monte Carlo runs. The probability region (with three angle biases for each sensor) is defined as

$$n_\omega = 3N_s, \tag{45}$$

$$r_1 = \chi^2_{(n_{\text{MC}}n_\omega)}(0.025), \tag{46}$$

$$r_2 = \chi^2_{(n_{\text{MC}}n_\omega)}(0.975), \tag{47}$$

$$\bar{\varepsilon} = \frac{1}{n_{\text{MC}}}\sum_{i=1}^{n_{\text{MC}}}\varepsilon^i, \tag{48}$$

$$\bar{\varepsilon} \in \left[\frac{r_1}{n_{\text{MC}}}, \frac{r_2}{n_{\text{MC}}}\right]. \tag{49}$$

In this case, with 4 sensors and 3 biases in each sensor, there are 12 parameters and the NEES should be around 12.

VI. SIMULATIONS AND RESULTS

A. Simulation Parameters

In order to evaluate the performance of this method, it is necessary to create a simulation of appropriate realism. To accomplish this, two scenarios are created, the

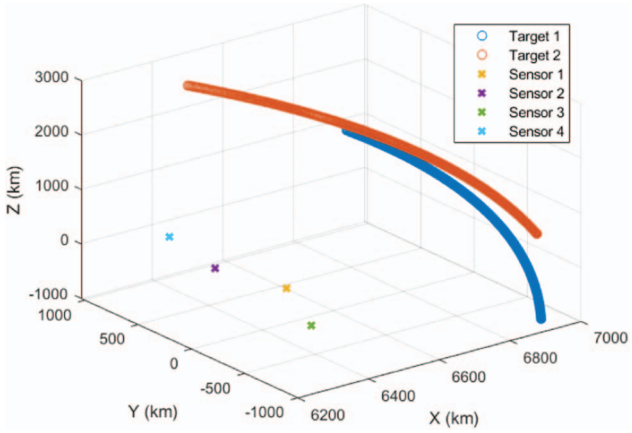


Fig. 2. Long-range sensor and target setup in ECEF coordinates. $K = 350$ s.

first being a long-distance orbital scenario and the second being a short-range maneuvering scenario. In the long-range scenario, there are four fixed sensors positioned near the equator at sea level observing two targets orbiting the Earth in a deterministic way. This scenario is useful because it is a baseline for performance in a deterministic motion scenario, which can be then compared to simultaneous target state and sensor bias estimation. In the short-range scenario, there are four ground-based sensors observing several targets that are moving toward a position on the ground with mid-air maneuvers. The reason for the short-range scenario is to show the ability of this method to estimate biases despite the difficulties in tracking a highly maneuvering target. These scenarios are shown in Figs. 2 and 3. The sensors have measurement noise standard deviation of 1 mrad and biases of 1 mrad. In the long-range scenario, the sensors take one measurement per second over 350 s ($K = 350$) and 100 Monte Carlo runs are used. In the short-range scenario, the sensors take 10 measurements per second over 40 s ($K = 400$) and 100 Monte Carlo runs are used.

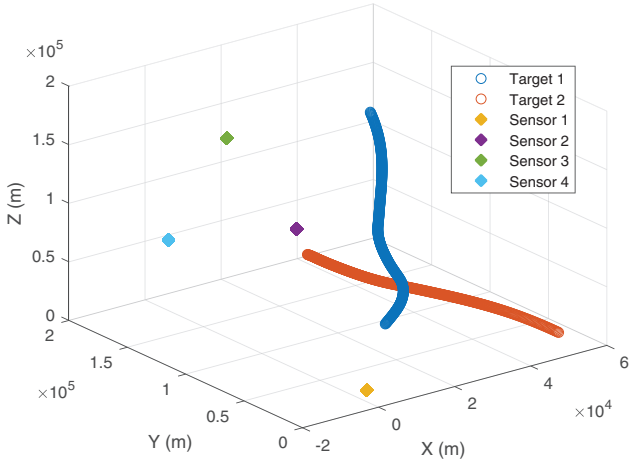


Fig. 3. Short-range sensor and target setup in ECEF coordinates. $K = 400$ s.

B. Statistical Efficiency

The CRLB is the lower bound on the variance of this estimator, meaning that the RMSE must be comparable to the square root of the CRLB. If the RMSE is accepted as equal (via a statistical hypothesis test) to the CRLB square root, then the estimator is considered statistically efficient. The simulations are first made to verify that this is the case for the estimator. Furthermore, the results are compared to the method previously resulted in [19] and the hybrid CRLB (HCRLB), an additional metric proposed in [13] and [14]. The HCRLB refers to the CRLB of joint target/bias estimation based on the original measurements, and hence can be considered the true lower bound. Since here we digest the original measurements into pseudo-measurements, there is potential loss of information, implying concomitant increase of the CRLB beyond the HCRLB. Additionally, for the long-range scenario, this method is compared to a previously developed method [19] that includes only the Cartesian positions from the conversion via triangulation, to show that using this method results in a lower CRLB as the conversion has not lost information about the biases. For the long-range scenario, the results are seen in Table I. We can see that for this scenario the new method is efficient and capable of estimating the biases with an error that is significantly lower than the noise standard deviation (1 mrad). The RMSE lies within the probability interval for all biases and the RMSE is less than 40% of the noise standard deviation for all biases.

Perhaps of even more interest, the method shown in the present work achieves the HCRLB, while the previous method [13], [14], [19] fails to do so. This means that no information about the biases is lost in converting the coordinates and no information can be added by using additional transformations and combinations of pseudo-measurements (such as using both CPA and triangulation[2]). For the short-range scenario, the results are seen in Table II, and similar conclusions can be drawn: the new method achieves efficiency even in the case of a maneuvering target. The reason why no information is lost is that the useful data related to the target position are included in the LOS angle measurements, which are incorporated into the pseudo-measurements. This is further related to the use of ILS, as during each iteration the LOS angles are updated to prevent error as the bias estimates iteratively update. It is not necessary to estimate the Cartesian position.

C. CRLB Relative to Number of Time Steps

The previous simulation results showed that the new method is efficient and capable of achieving strong bias estimates in favorable conditions. However, it is important to understand how much data may be necessary to

[2]This can be loosely compared to counting one's money forward and backward (à la dynamic programming) and adding the two.

Table I
Long-Range Scenario: Verification of the Statistical Efficiency with the CRLB, $n_{MC} = 100$ Runs; All Quantities are in mrad

Component	CRLB square root (present work)	Triangulation CRLB [19] square root	HCRLB [13], [14] square root	RMSE (present work)	95% probability interval (41)
Sensor 1 yaw bias	0.2203	0.3357	0.2203	0.2371	0.1902 0.2504
Sensor 1 pitch bias	0.3723	0.5033	0.3722	0.3530	0.3332 0.4115
Sensor 1 roll bias	0.1474	0.3053	0.1474	0.1578	0.1280 0.1667
Sensor 2 yaw bias	0.1563	0.2012	0.1563	0.1775	0.1334 0.1792
Sensor 2 pitch bias	0.3345	0.4280	0.3343	0.3309	0.2964 0.3726
Sensor 2 roll bias	0.0990	0.2729	0.0990	0.1094	0.0864 0.1116
Sensor 3 yaw bias	0.2019	0.3197	0.2019	0.2104	0.1753 0.2285
Sensor 3 pitch bias	0.3393	0.4609	0.3393	0.3187	0.3018 0.3769
Sensor 3 roll bias	0.1522	0.2928	0.1521	0.1600	0.1333 0.1710
Sensor 4 yaw bias	0.1046	0.1674	0.1045	0.1019	0.0929 0.1163
Sensor 4 pitch bias	0.3921	0.5214	0.3919	0.3821	0.3471 0.4372
Sensor 4 roll bias	0.1171	0.2705	0.1171	0.1282	0.1017 0.1324
		Average NEES 12.365	Chi-square 95% interval 11.059 12.979		

Table II
Short-Range Scenario: Verification of the Statistical Efficiency with the CRLB, $n_{MC} = 100$ runs; All Quantities are in mrad

Component	CRL square root (present work)	Triangulation CRLB [19] square root	HCRLB [13], [14] Square root	RMSE (present work)	95% probability interval (41)
Sensor 1 yaw bias	0.1306	0.1425	0.1305	0.1325	0.1158 0.1454
Sensor 1 pitch bias	0.1569	0.1750	0.1569	0.1629	0.1391 0.1746
Sensor 1 roll bias	0.0693	0.1203	0.0693	0.0706	0.0607 0.0779
Sensor 2 yaw bias	0.1092	0.1262	0.1092	0.1057	0.0963 0.1222
Sensor 2 pitch bias	0.1067	0.1385	0.1067	0.1162	0.0942 0.1193
Sensor 2 roll bias	0.0792	0.0955	0.0792	0.0722	0.0686 0.0897
Sensor 3 yaw bias	0.1319	0.1595	0.1319	0.1337	0.1160 0.1477
Sensor 3 pitch bias	0.0873	0.1019	0.0872	0.0977	0.0765 0.0978
Sensor 3 roll bias	0.0623	0.0632	0.0623	0.0595	0.0548 0.0697
Sensor 4 yaw bias	0.0799	0.0935	0.0799	0.0845	0.0697 0.0902
Sensor 4 pitch bias	0.1447	0.1676	0.1447	0.1552	0.1264 0.1631
Sensor 4 roll bias	0.0699	0.1148	0.0699	0.0659	0.0616 0.0782
		Average NEES 12.253	Chi-square 95% interval 11.059 12.979		

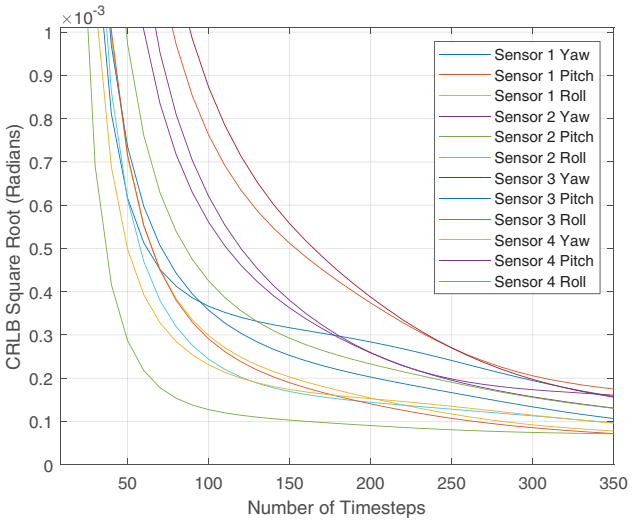


Fig. 4. CRLB square root of bias estimates compared to number of time steps for the short-range scenario with two targets.

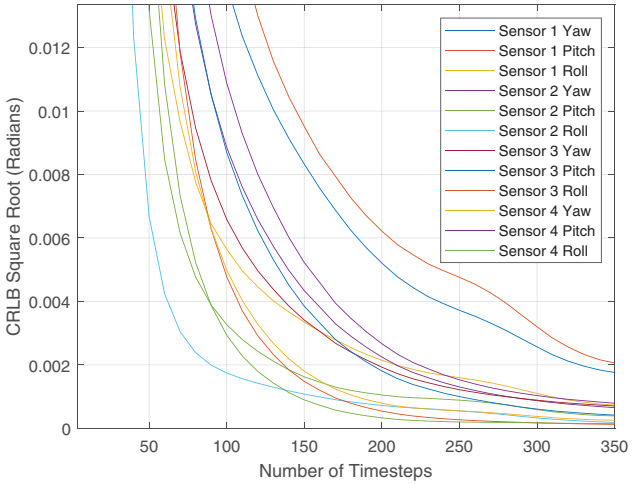


Fig. 6. CRLB square root of bias estimates compared to number of time steps for the short-range scenario with one target.

have a good bias estimate and what to expect in bad conditions. The CRLB is calculated for the short-range scenario but with a spread of time steps from 10 time steps (at 10 Hz, i.e., 1 s) to 400 time steps (40 s). The results of this are seen in Figs. 4 and 5.

In the case of this two-target short-range scenario, we see that within 150 time steps (15 s) all the bias errors reduce to below half of the noise standard deviation. This is particularly good as the bias estimation is able to overcome the bias error relatively quickly, and certainly before the targets reach their destination. Furthermore, the RMSE graph matches the CRLB graph, showing that this method retains efficiency even as the number of measurements decreases, which would accordingly reduce the observability and accuracy of bias estimation. This result proves a degree of resilience of this method to poor observability, as the method remains efficient, even when the error in the bias estimates is likely worse than the biases themselves. However, in this case there are two targets, and hence there is a more diverse set of data for elimination of the biases.

Next, we investigate the perhaps more common situation that there be only a single target. Figs. 6 and 7 show that the performance deteriorates. The CRLB of the bias estimates does not reduce to below the noise standard deviation until around 300 time steps (30 s) and two of the biases are significantly higher as a result of the sensor's position relative to the target's motion. The RMSE remains comparable to the CRLB even as the error increases significantly higher than the uncorrected bias error, proving efficiency in poor observability scenarios. Furthermore, we see that having two targets is better than having twice as much time, as seen by the CRLB being lower for two targets (Fig. 4) at 150 time steps than one target (Fig. 6) at 300 time steps. The biases affect the targets in Cartesian space differently as a result of their positions; therefore, the accuracy is improved greatly as a result of having additional targets.

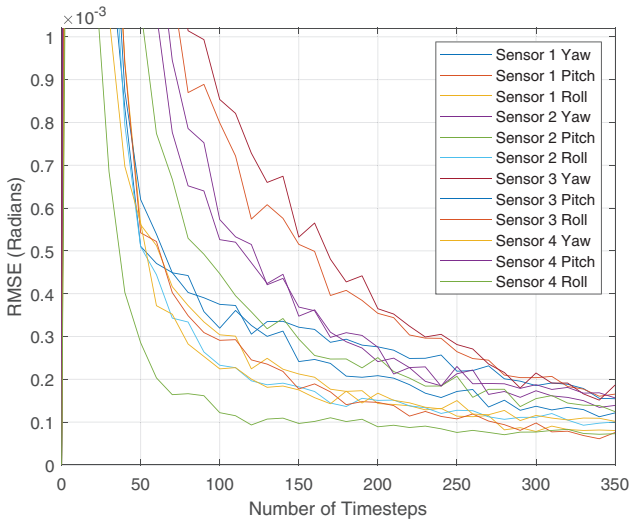


Fig. 5. RMSE of bias estimates compared to number of time steps for the short-range scenario with two targets.

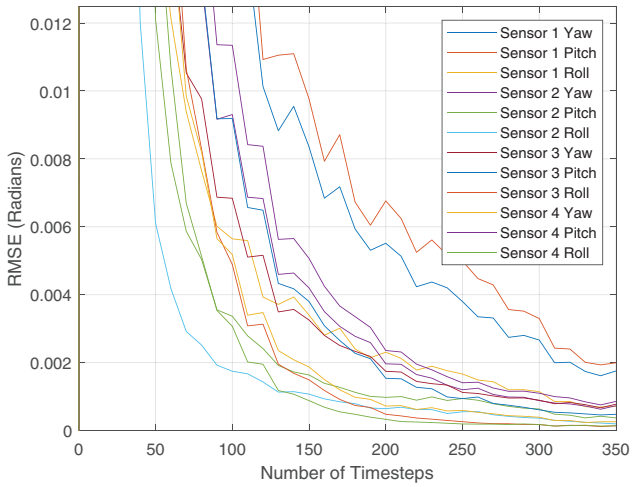


Fig. 7. CRLB square root of bias estimates compared to number of time steps for the short-range scenario with one target.

As it is impossible to achieve more accurate bias estimates than the CRLB, it may be necessary to either have knowledge of the target's state or include additional targets, such as friendly ones and known objects that are observed by the sensor, to improve bias estimates within a shorter time frame. Methods of including such "stationary emitters" are included in works such as [10] and [11].

VII. CONCLUSION

The CPA-based method is an effective tool for bias estimation in passive sensor data fusion applications. The bias pseudo-measurement method can be applied to angle-only sensors in 3-D to estimate the biases without target state estimation. The bias estimation CRLB is attained using this method and can be informative about whether the system has enough data to perform bias estimation or whether it is necessary to include additional information to improve accuracy. Furthermore, it is possible to reduce the bias residual error to significantly below the noise standard deviation. The simulations show that having additional targets improves bias estimation accuracy more than having a corresponding increase in time steps, meaning a more diverse set of measurements is better than simply having more.

APPENDIX A

The Jacobians used in this paper need to be calculated in order to convert the measurements and use ILS. We first specify the problem formulation that is used for our measurements and parameters. Before we can produce our bias pseudo-measurements, the effect of the nominal rotation must be accounted for and removed from the measurements. We use the transformation (13), which is expanded as

$$\begin{aligned}\xi_s^{t,b}(k) &= \begin{bmatrix}\alpha_s^{t,b}(k)\\ \epsilon_s^{t,b}(k)\end{bmatrix}\\ &= \begin{bmatrix}\tan^{-1}\left(\dfrac{\boldsymbol{\lambda}_s^{t,b,y}(k)}{\boldsymbol{\lambda}_s^{t,b,x}(k)}\right)\\ \tan^{-1}\left(\dfrac{\boldsymbol{\lambda}_s^{t,b,z}(k)}{\sqrt{\boldsymbol{\lambda}_s^{t,b,x}(k)^2+\boldsymbol{\lambda}_s^{t,b,y}(k)^2}}\right)\end{bmatrix}\\ &= h_s^i\left(\begin{bmatrix}\alpha_s^{t,n,b}(k)\\ \epsilon_s^{t,n,b}(k)\end{bmatrix}, \boldsymbol{\omega}_s^n\right), \end{aligned} \tag{A1}$$

where the variable $\boldsymbol{\lambda}_s^{t,b}$ is the LOS ray in common Cartesian space rotated by the bias rotation. To acquire it, the original LOS ray must be rotated by the inverse of the nominal rotation. This inverted rotation is

$$T(\boldsymbol{\omega}_s^n)^{-1} = \begin{bmatrix}T_{11,i}(\boldsymbol{\omega}_s^n) & T_{12,i}(\boldsymbol{\omega}_s^n) & T_{13,i}(\boldsymbol{\omega}_s^n)\\ T_{21,i}(\boldsymbol{\omega}_s^n) & T_{22,i}(\boldsymbol{\omega}_s^n) & T_{23,i}(\boldsymbol{\omega}_s^n)\\ T_{31,i}(\boldsymbol{\omega}_s^n) & T_{32,i}(\boldsymbol{\omega}_s^n) & T_{33,i}(\boldsymbol{\omega}_s^n)\end{bmatrix}. \tag{A2}$$

The LOS rays under rotations in common Cartesian space are

$$\begin{aligned}\boldsymbol{\lambda}_s^{t,n,b}(k) &= \begin{bmatrix}\boldsymbol{\lambda}_s^{t,n,b,x}(k)\\ \boldsymbol{\lambda}_s^{t,n,b,y}(k)\\ \boldsymbol{\lambda}_s^{t,n,b,z}(k)\end{bmatrix}\\ &= \begin{bmatrix}\cos(\alpha_s^{t,n,b}(k))\cos(\epsilon_s^{t,n,b}(k))\\ \sin(\alpha_s^{t,n,b}(k))\cos(\epsilon_s^{t,n,b}(k))\\ \sin(\epsilon_s^{t,n,b}(k))\end{bmatrix}, \end{aligned} \tag{A3}$$

$$\begin{aligned}\boldsymbol{\lambda}_s^{t,b}(k) &= \begin{bmatrix}\boldsymbol{\lambda}_s^{t,b,x}(k)\\ \boldsymbol{\lambda}_s^{t,b,y}(k)\\ \boldsymbol{\lambda}_s^{t,b,z}(k)\end{bmatrix}\\ &= \begin{bmatrix}\cos(\alpha_s^{t,b}(k))\cos(\epsilon_s^{t,b}(k))\\ \sin(\alpha_s^{t,b}(k))\cos(\epsilon_s^{t,b}(k))\\ \sin(\epsilon_s^{t,b}(k))\end{bmatrix}\\ &= T(\boldsymbol{\omega}_s^n)^{-1}\boldsymbol{\lambda}_s^{t,n,b}(k). \end{aligned} \tag{A4}$$

The equations for the individual Cartesian components of the ray rotated by the inverse nominal rotation are

$$\begin{aligned}\boldsymbol{\lambda}_s^{t,b,x}(k) &= T_{11,i}(\boldsymbol{\omega}_s^n)\boldsymbol{\lambda}_s^{t,n,b,x}(k) + T_{12,i}(\boldsymbol{\omega}_s^n)\boldsymbol{\lambda}_s^{t,n,b,y}(k)\\ &\quad + T_{13,i}(\boldsymbol{\omega}_s^n)\boldsymbol{\lambda}_s^{t,n,b,z}(k), \end{aligned} \tag{A5}$$

$$\begin{aligned}\boldsymbol{\lambda}_s^{t,b,y}(k) &= T_{21,i}(\boldsymbol{\omega}_s^n)\boldsymbol{\lambda}_s^{t,n,b,x}(k) + T_{22,i}(\boldsymbol{\omega}_s^n)\boldsymbol{\lambda}_s^{t,n,b,y}(k)\\ &\quad + T_{23,i}(\boldsymbol{\omega}_s^n)\boldsymbol{\lambda}_s^{t,n,b,z}(k), \end{aligned} \tag{A6}$$

$$\begin{aligned}\boldsymbol{\lambda}_s^{t,b,z}(k) &= T_{31,i}(\boldsymbol{\omega}_s^n)\boldsymbol{\lambda}_s^{t,n,b,x}(k) + T_{32,i}(\boldsymbol{\omega}_s^n)\boldsymbol{\lambda}_s^{t,n,b,y}(k)\\ &\quad + T_{33,i}(\boldsymbol{\omega}_s^n)\boldsymbol{\lambda}_s^{t,n,b,z}(k). \end{aligned} \tag{A7}$$

These equations are used in the calculation of the individual partial derivatives for the Jacobian D from equation (11), which are

$$\begin{aligned}\frac{\partial\alpha_s^{t,b}(k)}{\partial\alpha_s^{t,n,b}(k)} &= \left(T_{11,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,x}(k)} + T_{21,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,y}(k)}\right.\\ &\quad \left. + T_{31,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,z}(k)}\right)\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,x}(k)}{\partial\alpha_s^{t,n,b}(k)}\\ &\quad + \left(T_{12,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,x}(k)} + T_{22,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,y}(k)}\right.\\ &\quad \left. + T_{32,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,z}(k)}\right)\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,y}(k)}{\partial\alpha_s^{t,n,b}(k)} \end{aligned}$$

$$+\left(T_{13,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,x}(k)}+T_{23,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,y}(k)}+T_{33,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,z}(k)}\right)\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,z}(k)}{\partial\alpha_s^{t,n,b}(k)}, \quad \text{(A8)}$$

$$\frac{\partial\alpha_s^{t,b}(k)}{\partial\epsilon_s^{t,n,b}(k)}=\left(T_{11,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,x}(k)}+T_{21,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,y}(k)}+T_{31,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,z}(k)}\right)\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,x}(k)}{\partial\epsilon_s^{t,n,b}(k)}$$
$$+\left(T_{12,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,x}(k)}+T_{22,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,y}(k)}+T_{32,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,z}(k)}\right)\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,y}(k)}{\partial\epsilon_s^{t,n,b}(k)}$$
$$+\left(T_{13,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,x}(k)}+T_{23,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,y}(k)}+T_{33,i}(\boldsymbol{\omega}_s^n)\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,z}(k)}\right)\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,z}(k)}{\partial\epsilon_s^{t,n,b}(k)}, \quad \text{(A9)}$$

$$\frac{\partial\epsilon_s^{t,b}(k)}{\partial\alpha_s^{t,n,b}(k)}=\left(T_{11,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,x}(k)}+T_{21,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,y}(k)}+T_{31,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,z}(k)}\right)\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,x}(k)}{\partial\alpha_s^{t,n,b}(k)}$$
$$+\left(T_{12,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,x}(k)}+T_{22,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,y}(k)}+T_{32,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,z}(k)}\right)\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,y}(k)}{\partial\alpha_s^{t,n,b}(k)}$$
$$+\left(T_{13,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,x}(k)}+T_{23,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,y}(k)}+T_{33,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,z}(k)}\right)\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,z}(k)}{\partial\alpha_s^{t,n,b}(k)}, \quad \text{(A10)}$$

$$\frac{\partial\epsilon_s^{t,b}(k)}{\partial\epsilon_s^{t,n,b}(k)}=\left(T_{11,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,x}(k)}+T_{21,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,y}(k)}+T_{31,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,z}(k)}\right)\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,x}(k)}{\partial\epsilon_s^{t,n,b}(k)}$$
$$+\left(T_{12,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,x}(k)}+T_{22,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,y}(k)}+T_{32,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,z}(k)}\right)\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,y}(k)}{\partial\epsilon_s^{t,n,b}(k)}$$
$$+\left(T_{13,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,x}(k)}+T_{23,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,y}(k)}+T_{33,i}(\boldsymbol{\omega}_s^n)\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,z}(k)}\right)\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,z}(k)}{\partial\epsilon_s^{t,n,b}(k)}, \quad \text{(A11)}$$

$$\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,x}(k)}=\frac{-\boldsymbol{\lambda}_s^{t,b,y}(k)}{\boldsymbol{\lambda}_s^{t,b,x}(k)^2+\boldsymbol{\lambda}_s^{t,b,y}(k)^2}, \quad \text{(A12)}$$

$$\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,y}(k)}=\frac{\boldsymbol{\lambda}_s^{t,b,x}(k)}{\boldsymbol{\lambda}_s^{t,b,x}(k)^2+\boldsymbol{\lambda}_s^{t,b,y}(k)^2}, \quad \text{(A13)}$$

$$\frac{\partial\alpha_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,z}(k)}=0, \quad \text{(A14)}$$

$$\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,x}(k)}=\frac{-\boldsymbol{\lambda}_s^{t,b,x}(k)\boldsymbol{\lambda}_s^{t,b,z}(k)}{\sqrt{\boldsymbol{\lambda}_s^{t,b,x}(k)^2+\boldsymbol{\lambda}_s^{t,b,y}(k)^2}||\boldsymbol{\lambda}_s^{t,b}(k)||^2}, \quad \text{(A15)}$$

$$\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,y}(k)}=\frac{-\boldsymbol{\lambda}_s^{t,b,y}(k)\boldsymbol{\lambda}_s^{t,b,z}(k)}{\sqrt{\boldsymbol{\lambda}_s^{t,b,x}(k)^2+\boldsymbol{\lambda}_s^{t,b,y}(k)^2}||\boldsymbol{\lambda}_s^{t,b}(k)||^2}, \quad \text{(A16)}$$

$$\frac{\partial\epsilon_s^{t,b}(k)}{\partial\boldsymbol{\lambda}_s^{t,b,z}(k)}=\frac{\sqrt{\boldsymbol{\lambda}_s^{t,b,x}(k)^2+\boldsymbol{\lambda}_s^{t,b,y}(k)^2}}{(\boldsymbol{\lambda}_s^{t,b,x}(k)^2+\boldsymbol{\lambda}_s^{t,b,y}(k)^2+\boldsymbol{\lambda}_s^{t,b,z}(k)^2)}, \quad \text{(A17)}$$

$$\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,x}(k)}{\partial\alpha_s^{t,n,b}(k)}=-\sin(\alpha_s^{t,n,b}(k))\cos(\epsilon_s^{t,n,b}(k)), \quad \text{(A18)}$$

$$\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,x}(k)}{\partial\epsilon_s^{t,n,b}(k)}=-\cos(\alpha_s^{t,n,b}(k))\sin(\epsilon_s^{t,n,b}(k)), \quad \text{(A19)}$$

$$\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,y}(k)}{\partial\alpha_s^{t,n,b}(k)}=\cos(\alpha_s^{t,n,b}(k))\cos(\epsilon_s^{t,n,b}(k)), \quad \text{(A20)}$$

$$\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,y}(k)}{\partial\epsilon_s^{t,n,b}(k)}=-\sin(\alpha_s^{t,n,b}(k))\sin(\epsilon_s^{t,n,b}(k)), \quad \text{(A21)}$$

$$\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,z}(k)}{\partial\alpha_s^{t,n,b}(k)}=0, \quad \text{(A22)}$$

$$\frac{\partial\boldsymbol{\lambda}_s^{t,n,b,z}(k)}{\partial\epsilon_s^{t,n,b}(k)}=\cos(\epsilon_s^{t,n,b}(k)). \quad \text{(A23)}$$

With D calculated, the next step is representing the effects of the biases on the azimuth and elevation measurements using the Jacobian C.

$$C_s^{t,b}(k)=\begin{bmatrix}\frac{\partial\alpha_s^{t,b}(k)}{\partial\theta_s} & \frac{\partial\alpha_s^{t,b}(k)}{\partial\phi_s} & \frac{\partial\alpha_s^{t,b}(k)}{\partial\psi_s}\\ \frac{\partial\epsilon_s^{t,b}(k)}{\partial\theta_s} & \frac{\partial\epsilon_s^{t,b}(k)}{\partial\phi_s} & \frac{\partial\epsilon_s^{t,b}(k)}{\partial\psi_s}\end{bmatrix}, \quad \text{(A24)}$$

$$\boldsymbol{\lambda}_s^t(k) = \begin{bmatrix} \boldsymbol{\lambda}_s^{t,x}(k) \\ \boldsymbol{\lambda}_s^{t,y}(k) \\ \boldsymbol{\lambda}_s^{t,z}(k) \end{bmatrix} = \begin{bmatrix} \cos(\alpha_s^t(k))\cos(\epsilon_s^t(k)) \\ \sin(\alpha_s^t(k))\cos(\epsilon_s^t(k)) \\ \sin(\epsilon_s^t(k)) \end{bmatrix} = T(\boldsymbol{\omega}_s^b)^{-1}\boldsymbol{\lambda}_s^{t,b}(k), \tag{A25}$$

$$\boldsymbol{\lambda}_s^{t,b,x}(k) = T_{11}(\boldsymbol{\omega}_s^b)\boldsymbol{\lambda}_s^{t,x}(k) + T_{12}(\boldsymbol{\omega}_s^b)\boldsymbol{\lambda}_s^{t,y}(k) + T_{13}(\boldsymbol{\omega}_s^b)\boldsymbol{\lambda}_s^{t,z}(k), \tag{A26}$$

$$\boldsymbol{\lambda}_s^{t,b,y}(k) = T_{21}(\boldsymbol{\omega}_s^b)\boldsymbol{\lambda}_s^{t,x}(k) + T_{22}(\boldsymbol{\omega}_s^b)\boldsymbol{\lambda}_s^{t,y}(k) + T_{23}(\boldsymbol{\omega}_s^b)\boldsymbol{\lambda}_s^{t,z}(k), \tag{A27}$$

$$\boldsymbol{\lambda}_s^{t,b,z}(k) = T_{31}(\boldsymbol{\omega}_s^b)\boldsymbol{\lambda}_s^{t,x}(k) + T_{32}(\boldsymbol{\omega}_s^b)\boldsymbol{\lambda}_s^{t,y}(k) + T_{33}(\boldsymbol{\omega}_s^b)\boldsymbol{\lambda}_s^{t,z}(k), \tag{A28}$$

$$\begin{aligned} \frac{\partial \alpha_s^{t,b}(k)}{\partial \boldsymbol{\omega}_s^b(i)} = & \left(\frac{\partial \alpha_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,x}(k)} \frac{\partial T_{11}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} + \frac{\partial \alpha_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,y}(k)} \frac{\partial T_{21}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} \right. \\ & \left. + \frac{\partial \alpha_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,z}(k)} \frac{\partial T_{31}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} \right) \boldsymbol{\lambda}_s^{t,x}(k) \\ & + \left(\frac{\partial \alpha_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,x}(k)} \frac{\partial T_{12}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} + \frac{\partial \alpha_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,y}(k)} \frac{\partial T_{22}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} \right. \\ & \left. + \frac{\partial \alpha_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,z}(k)} \frac{\partial T_{32}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} \right) \boldsymbol{\lambda}_s^{t,y}(k) \\ & + \left(\frac{\partial \alpha_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,x}(k)} \frac{\partial T_{13}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} + \frac{\partial \alpha_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,y}(k)} \frac{\partial T_{23}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} \right. \\ & \left. + \frac{\partial \alpha_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,z}(k)} \frac{\partial T_{33}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} \right) \boldsymbol{\lambda}_s^{t,z}(k), \end{aligned} \tag{A29}$$

$$\begin{aligned} \frac{\partial \epsilon_s^{t,b}(k)}{\partial \boldsymbol{\omega}_s^b(i)} = & \left(\frac{\partial \epsilon_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,x}(k)} \frac{\partial T_{11}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} + \frac{\partial \epsilon_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,y}(k)} \frac{\partial T_{21}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} \right. \\ & \left. + \frac{\partial \epsilon_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,z}(k)} \frac{\partial T_{31}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} \right) \boldsymbol{\lambda}_s^{t,x}(k) \\ & + \left(\frac{\partial \epsilon_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,x}(k)} \frac{\partial T_{12}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} + \frac{\partial \epsilon_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,y}(k)} \frac{\partial T_{22}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} \right. \\ & \left. + \frac{\partial \epsilon_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,z}(k)} \frac{\partial T_{32}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} \right) \boldsymbol{\lambda}_s^{t,y}(k) \\ & + \left(\frac{\partial \epsilon_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,x}(k)} \frac{\partial T_{13}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} + \frac{\partial \epsilon_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,y}(k)} \frac{\partial T_{23}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} \right. \\ & \left. + \frac{\partial \epsilon_s^{t,b}(k)}{\partial \boldsymbol{\lambda}_s^{t,b,z}(k)} \frac{\partial T_{33}(\boldsymbol{\omega}_s^b)}{\partial \boldsymbol{\omega}_s^b(i)} \right) \boldsymbol{\lambda}_s^{t,z}(k). \end{aligned} \tag{A30}$$

To calculate C, it is necessary to evaluate how the biases affect the azimuth and elevation measurements, which requires knowledge of the unbiased azimuth and elevation measurements. To do this, we can debias our measurements using the same method as before with the nominal rotation based on the current bias estimate.

$$\xi_s^t(k) \approx \hat{\xi}_s^{t,j}(k) = \begin{bmatrix} \hat{\alpha}_s^{t,j}(k) \\ \hat{\epsilon}_s^{t,j}(k) \end{bmatrix} = h^i\left(\zeta_s^{t,b}(k), \hat{\boldsymbol{\omega}}_s^{b,j}\right), \tag{A31}$$

$$\frac{\partial T_{11}(\boldsymbol{\omega}_s^b)}{\partial \theta_s^b} = -\sin(\theta_s^b)\cos(\phi_s^b), \tag{A32}$$

$$\frac{\partial T_{11}(\boldsymbol{\omega}_s^b)}{\partial \phi_s^b} = -\cos(\theta_s^b)\sin(\phi_s^b), \tag{A33}$$

$$\frac{\partial T_{11}(\boldsymbol{\omega}_s^b)}{\partial \psi_s^b} = 0, \tag{A34}$$

$$\frac{\partial T_{12}(\boldsymbol{\omega}_s^b)}{\partial \theta_s^b} = -\sin(\theta_s^b)\sin(\phi_s^b)\sin(\psi_s^b) - \cos(\theta_s^b)\cos(\psi_s^b), \tag{A35}$$

$$\frac{\partial T_{12}(\boldsymbol{\omega}_s^b)}{\partial \phi_s^b} = \cos(\theta_s^b)\cos(\phi_s^b)\sin(\psi_s^b) \tag{A36}$$

$$\frac{\partial T_{12}(\boldsymbol{\omega}_s^b)}{\partial \psi_s^b} = \cos(\theta_s^b)\sin(\phi_s^b)\cos(\psi_s^b) + \sin(\theta_s^b)\sin(\psi_s^b), \tag{A37}$$

$$\frac{\partial T_{13}(\boldsymbol{\omega}_s^b)}{\partial \theta_s^b} = -\sin(\theta_s^b)\sin(\phi_s^b)\cos(\psi_s^b) + \cos(\theta_s^b)\sin(\psi_s^b), \tag{A38}$$

$$\frac{\partial T_{13}(\boldsymbol{\omega}_s^b)}{\partial \phi_s^b} = \cos(\theta_s^b)\cos(\phi_s^b)\cos(\psi_s^b), \tag{A39}$$

$$\frac{\partial T_{13}(\boldsymbol{\omega}_s^b)}{\partial \psi_s^b} = -\cos(\theta_s^b)\sin(\phi_s^b)\sin(\psi_s^b) + \sin(\theta_s^b)\cos(\psi_s^b), \tag{A40}$$

$$\frac{\partial T_{21}(\boldsymbol{\omega}_s^b)}{\partial \theta_s^b} = \cos(\theta_s^b)\cos(\phi_s^b), \tag{A41}$$

$$\frac{\partial T_{21}(\boldsymbol{\omega}_s^b)}{\partial \phi_s^b} = -\sin(\theta_s^b)\sin(\phi_s^b), \tag{A42}$$

$$\frac{\partial T_{21}(\boldsymbol{\omega}_s^b)}{\partial \psi_s^b} = 0, \tag{A43}$$

$$\frac{\partial T_{22}(\boldsymbol{\omega}_s^b)}{\partial \theta_s^b} = \cos(\theta_s^b)\sin(\phi_s^b)\sin(\psi_s^b) - \sin(\theta_s^b)\cos(\psi_s^b), \tag{A44}$$

$$\frac{\partial T_{22}(\boldsymbol{\omega}_s^b)}{\partial \phi_s^b} = \sin(\theta_s^b)\cos(\phi_s^b)\sin(\psi_s^b), \tag{A45}$$

$$\frac{\partial T_{22}(\boldsymbol{\omega}_s^b)}{\partial \psi_s^b} = \sin(\theta_s^b)\sin(\phi_s^b)\cos(\psi_s^b) - \cos(\theta_s^b)\sin(\psi_s^b), \tag{A46}$$

$$\frac{\partial T_{23}(\boldsymbol{\omega}_s^b)}{\partial \theta_s^b} = \cos(\theta_s^b)\sin(\phi_s^b)\cos(\psi_s^b) + \sin(\theta_s^b)\sin(\psi_s^b), \tag{A47}$$

$$\frac{\partial T_{23}(\boldsymbol{\omega}_s^b)}{\partial \phi_s^b} = \sin(\theta_s^b)\cos(\phi_s^b)\cos(\psi_s^b), \tag{A48}$$

$$\frac{\partial T_{23}(\boldsymbol{\omega}_s^b)}{\partial \psi_s^b} = -\sin(\theta_s^b)\sin(\phi_s^b)\sin(\psi_s^b) - \cos(\theta_s^b)\cos(\psi_s^b), \tag{A49}$$

$$\frac{\partial T_{31}(\boldsymbol{\omega}_s^b)}{\partial \theta_s^b} = 0, \tag{A50}$$

$$\frac{\partial T_{31}(\boldsymbol{\omega}_s^b)}{\partial \phi_s^b} = -\cos(\phi_s^b), \tag{A51}$$

$$\frac{\partial T_{31}(\boldsymbol{\omega}_s^b)}{\partial \psi_s^b} = 0, \tag{A52}$$

$$\frac{\partial T_{32}(\boldsymbol{\omega}_s^b)}{\partial \theta_s^b} = 0, \tag{A53}$$

$$\frac{\partial T_{32}(\boldsymbol{\omega}_s^b)}{\partial \phi_s^b} = -\sin(\phi_s^b)\sin(\psi_s^b), \tag{A54}$$

$$\frac{\partial T_{32}(\boldsymbol{\omega}_s^b)}{\partial \psi_s^b} = \cos(\phi_s^b)\cos(\psi_s^b), \tag{A55}$$

$$\frac{\partial T_{33}(\boldsymbol{\omega}_s^b)}{\partial \theta_s^b} = 0, \tag{A56}$$

$$\frac{\partial T_{33}(\boldsymbol{\omega}_s^b)}{\partial \phi_s^b} = -\sin(\phi_s^b)\cos(\psi_s^b), \tag{A57}$$

$$\frac{\partial T_{33}(\boldsymbol{\omega}_s^b)}{\partial \psi_s^b} = -\cos(\phi_s^b)\sin(\psi_s^b). \tag{A58}$$

In order to transform the effect of the biases when converting into Cartesian, we use the Jacobian B from equation (24) for which the gradients are calculated as

$$\nabla_{\xi_1^t,\xi_2^t}\boldsymbol{x}_{12'}^{t,c}(k) = \begin{bmatrix} \frac{\partial x_{12'}^{t,c}}{\partial \alpha_1^t} & \frac{\partial x_{12'}^{t,c}}{\partial \epsilon_1^t} & \frac{\partial x_{12'}^{t,c}}{\partial \alpha_2^t} & \frac{\partial x_{12'}^{t,c}}{\partial \epsilon_2^t} \\ \frac{\partial y_{12'}^{t,c}}{\partial \alpha_1^t} & \frac{\partial y_{12'}^{t,c}}{\partial \epsilon_1^t} & \frac{\partial y_{12'}^{t,c}}{\partial \alpha_2^t} & \frac{\partial y_{12'}^{t,c}}{\partial \epsilon_2^t} \\ \frac{\partial z_{12'}^{t,c}}{\partial \alpha_1^t} & \frac{\partial z_{12'}^{t,c}}{\partial \epsilon_1^t} & \frac{\partial z_{12'}^{t,c}}{\partial \alpha_2^t} & \frac{\partial z_{12'}^{t,c}}{\partial \epsilon_2^t} \end{bmatrix}, \tag{A59}$$

$$\nabla_{\xi_1^t,\xi_2^t}\boldsymbol{x}_{12''}^{t,c}(k) = \begin{bmatrix} \frac{\partial x_{12''}^{t,c}}{\partial \alpha_1^t} & \frac{\partial x_{12''}^{t,c}}{\partial \epsilon_1^t} & \frac{\partial x_{12''}^{t,c}}{\partial \alpha_2^t} & \frac{\partial x_{12''}^{t,c}}{\partial \epsilon_2^t} \\ \frac{\partial y_{12''}^{t,c}}{\partial \alpha_1^t} & \frac{\partial y_{12''}^{t,c}}{\partial \epsilon_1^t} & \frac{\partial y_{12''}^{t,c}}{\partial \alpha_2^t} & \frac{\partial y_{12''}^{t,c}}{\partial \epsilon_2^t} \\ \frac{\partial z_{12''}^{t,c}}{\partial \alpha_1^t} & \frac{\partial z_{12''}^{t,c}}{\partial \epsilon_1^t} & \frac{\partial z_{12''}^{t,c}}{\partial \alpha_2^t} & \frac{\partial z_{12''}^{t,c}}{\partial \epsilon_2^t} \end{bmatrix}. \tag{A60}$$

To improve clarity in the calculation of these partial derivatives, the conversion equations are simplified by using

$$\gamma_1^t = (\boldsymbol{\lambda}_1^t)'\mathbf{p}_{1,2} - ((\boldsymbol{\lambda}_1^t)'\boldsymbol{\lambda}_2^t)((\boldsymbol{\lambda}_2^t)'\mathbf{p}_{1,2}), \tag{A61}$$

$$\gamma_2^t = ((\boldsymbol{\lambda}_1^t)'\boldsymbol{\lambda}_2^t)((\boldsymbol{\lambda}_1^t)'\mathbf{p}_{1,2}) - (\boldsymbol{\lambda}_2^t)'\mathbf{p}_{1,2}, \tag{A62}$$

$$\gamma_{x,1}^t = \boldsymbol{\lambda}_1^{t,x}\gamma_1^t, \tag{A63}$$

$$\gamma_{y,1}^t = \boldsymbol{\lambda}_1^{t,y}\gamma_1^t, \tag{A64}$$

$$\gamma_{z,1}^t = \boldsymbol{\lambda}_1^{t,z}\gamma_1^t, \tag{A65}$$

$$\gamma_{x,2}^t = \boldsymbol{\lambda}_2^{t,x}\gamma_2^t, \tag{A66}$$

$$\gamma_{y,2}^t = \boldsymbol{\lambda}_2^{t,y}\gamma_2^t, \tag{A67}$$

$$\gamma_{z,2}^t = \boldsymbol{\lambda}_2^{t,z}\gamma_2^t, \tag{A68}$$

$$\kappa_{12}^t = 1 - ((\boldsymbol{\lambda}_1^t)'\boldsymbol{\lambda}_2^t)^2, \tag{A69}$$

which results in the equations

$$x_{12'}^{t,c} = x_1 + \frac{\gamma_{x,1}^t}{\kappa_{12}^t}, \tag{A70}$$

$$x_{12''}^{t,c} = x_2 + \frac{\gamma_{x,2}^t}{\kappa_{12}^t}, \tag{A71}$$

$$y_{12'}^{t,c} = y_1 + \frac{\gamma_{y,1}^t}{\kappa_{12}^t}, \tag{A72}$$

$$y_{12''}^{t,c} = y_2 + \frac{\gamma_{y,2}^t}{\kappa_{12}^t}, \tag{A73}$$

$$z_{12'}^{t,c} = z_1 + \frac{\gamma_{z,1}^t}{\kappa_{12}^t}, \tag{A74}$$

$$z_{12''}^{t,c} = z_2 + \frac{\gamma_{z,2}^t}{\kappa_{12}^t}, \tag{A75}$$

$$\frac{\partial x_{12'}^{t,c}}{\partial \xi_s^t} = \frac{\kappa_{12}^t \frac{\partial \gamma_{x,1}^t}{\partial \xi_s^t} - \gamma_{x,1}^t \frac{\partial \kappa_{12}^t}{\partial \xi_s^t}}{(\kappa_{12}^t)^2}, \tag{A76}$$

$$\frac{\partial x_{12''}^{t,c}}{\partial \xi_s^t} = \frac{\kappa_{12}^t \frac{\partial \gamma_{x,2}^t}{\partial \xi_s^t} - \gamma_{x,2}^t \frac{\partial \kappa_{12}^t}{\partial \xi_s^t}}{(\kappa_{12}^t)^2}, \tag{A77}$$

$$\frac{\partial y_{12'}^{t,c}}{\partial \xi_s^t} = \frac{\kappa_{12}^t \frac{\partial \gamma_{y,1}^t}{\partial \xi_s^t} - \gamma_{y,1}^t \frac{\partial \kappa_{12}^t}{\partial \xi_s^t}}{(\kappa_{12}^t)^2}, \tag{A78}$$

$$\frac{\partial y_{12''}^{t,c}}{\partial \xi_s^t} = \frac{\kappa_{12}^t \frac{\partial \gamma_{y,2}^t}{\partial \xi_s^t} - \gamma_{y,2}^t \frac{\partial \kappa_{12}^t}{\partial \xi_s^t}}{(\kappa_{12}^t)^2}, \tag{A79}$$

$$\frac{\partial z_{12'}^{t,c}}{\partial \xi_s^t} = \frac{\kappa_{12}^t \frac{\partial \gamma_{z,1}^t}{\partial \xi_s^t} - \gamma_{z,1}^t \frac{\partial \kappa_{12}^t}{\partial \xi_s^t}}{(\kappa_{12}^t)^2}, \tag{A80}$$

$$\frac{\partial z_{12''}^{t,c}}{\partial \xi_s^t} = \frac{\kappa_{12}^t \frac{\partial \gamma_{z,2}^t}{\partial \xi_s^t} - \gamma_{z,2}^t \frac{\partial \kappa_{12}^t}{\partial \xi_s^t}}{(\kappa_{12}^t)^2}, \tag{A81}$$

$$\frac{\partial \gamma_{x,1}^t}{\partial \alpha_1^t} = -\sin(\alpha_1^t)\cos(\epsilon_1^t)\gamma_1^t + \cos(\alpha_1^t)\cos(\epsilon_1^t)\frac{\partial \gamma_1^t}{\partial \alpha_1^t}, \tag{A82}$$

$$\frac{\partial \gamma_{x,1}^t}{\partial \alpha_2^t} = \cos(\alpha_1^t)\cos(\epsilon_1^t)\frac{\partial \gamma_1^t}{\partial \alpha_2^t}, \tag{A83}$$

$$\frac{\partial \gamma_{x,1}^t}{\partial \epsilon_1^t} = -\cos(\alpha_1^t)\sin(\epsilon_1^t)\gamma_1^t + \cos(\alpha_1^t)\cos(\epsilon_1^t)\frac{\partial \gamma_1^t}{\partial \epsilon_1^t}, \tag{A84}$$

$$\frac{\partial \gamma_{x,1}^t}{\partial \epsilon_2^t} = \cos(\alpha_1^t)\cos(\epsilon_1^t)\frac{\partial \gamma_1^t}{\partial \epsilon_2^t}, \tag{A85}$$

$$\frac{\partial \gamma_{y,1}^t}{\partial \alpha_1^t} = \cos(\alpha_1^t)\cos(\epsilon_1^t)\gamma_1^t + \sin(\alpha_1^t)\cos(\epsilon_1^t)\frac{\partial \gamma_1^t}{\partial \alpha_1^t}, \tag{A86}$$

$$\frac{\partial \gamma_{y,1}^t}{\partial \alpha_2^t} = \sin(\alpha_1^t)\cos(\epsilon_1^t)\frac{\partial \gamma_1^t}{\partial \alpha_2^t}, \tag{A87}$$

$$\frac{\partial \gamma_{y,1}^t}{\partial \epsilon_1^t} = -\sin(\alpha_1^t)\sin(\epsilon_1^t)\gamma_1^t + \sin(\alpha_1^t)\cos(\epsilon_1^t)\frac{\partial \gamma_1^t}{\partial \epsilon_1^t}, \tag{A88}$$

$$\frac{\partial \gamma_{y,1}^t}{\partial \epsilon_2^t} = \sin(\alpha_1^t)\cos(\epsilon_1^t)\frac{\partial \gamma_1^t}{\partial \epsilon_2^t}, \tag{A89}$$

$$\frac{\partial \gamma_{z,1}^t}{\partial \alpha_1^t} = \sin(\epsilon_1^t)\frac{\partial \gamma_1^t}{\partial \alpha_1^t}, \tag{A90}$$

$$\frac{\partial \gamma_{z,1}^t}{\partial \alpha_2^t} = \sin(\epsilon_1^t)\frac{\partial \gamma_1^t}{\partial \alpha_2^t}, \tag{A91}$$

$$\frac{\partial \gamma_{z,1}^t}{\partial \epsilon_1^t} = \cos(\epsilon_1^t)\gamma_1^t + \sin(\epsilon_1^t)\frac{\partial \gamma_1^t}{\partial \epsilon_1^t}, \tag{A92}$$

$$\frac{\partial \gamma_{z,1}^t}{\partial \epsilon_2^t} = \sin(\epsilon_1^t)\frac{\partial \gamma_1^t}{\partial \epsilon_2^t}, \tag{A93}$$

$$\frac{\partial \gamma_{x,2}^t}{\partial \alpha_1^t} = \cos(\alpha_2^t)\cos(\epsilon_2^t)\frac{\partial \gamma_2^t}{\partial \alpha_1^t}, \tag{A94}$$

$$\frac{\partial \gamma_{x,2}^t}{\partial \alpha_2^t} = -\sin(\alpha_2^t)\cos(\epsilon_2^t)\gamma_2^t + \cos(\alpha_2^t)\cos(\epsilon_2^t)\frac{\partial \gamma_2^t}{\partial \alpha_2^t}, \tag{A95}$$

$$\frac{\partial \gamma_{x,2}^t}{\partial \epsilon_1^t} = \cos(\alpha_2^t)\cos(\epsilon_2^t)\frac{\partial \gamma_2^t}{\partial \epsilon_1^t}, \tag{A96}$$

$$\frac{\partial \gamma_{x,2}^t}{\partial \epsilon_2^t} = -\cos(\alpha_2^t)\sin(\epsilon_2^t)\gamma_2^t + \cos(\alpha_2^t)\cos(\epsilon_2^t)\frac{\partial \gamma_2^t}{\partial \epsilon_2^t}, \tag{A97}$$

$$\frac{\partial \gamma_{y,2}^t}{\partial \alpha_1^t} = \sin(\alpha_2^t)\cos(\epsilon_2^t)\frac{\partial \gamma_2^t}{\partial \alpha_1^t}, \tag{A98}$$

$$\frac{\partial \gamma_{y,2}^t}{\partial \alpha_2^t} = \cos(\alpha_2^t)\cos(\epsilon_2^t)\gamma_2^t + \sin(\alpha_2^t)\cos(\epsilon_2^t)\frac{\partial \gamma_2^t}{\partial \alpha_2^t}, \tag{A99}$$

$$\frac{\partial \gamma_{y,2}^t}{\partial \epsilon_1^t} = \sin(\alpha_2^t)\cos(\epsilon_2^t)\frac{\partial \gamma_2^t}{\partial \epsilon_1^t}, \tag{A100}$$

$$\frac{\partial \gamma_{y,2}^t}{\partial \epsilon_2^t} = -\sin(\alpha_2^t)\sin(\epsilon_2^t)\gamma_2^t + \sin(\alpha_2^t)\cos(\epsilon_2^t)\frac{\partial \gamma_2^t}{\partial \epsilon_2^t}, \tag{A101}$$

$$\frac{\partial \gamma_{z,2}^t}{\partial \alpha_1^t} = \sin(\epsilon_2^t)\frac{\partial \gamma_2^t}{\partial \alpha_1^t}, \tag{A102}$$

$$\frac{\partial \gamma_{z,2}^t}{\partial \alpha_2^t} = \sin(\epsilon_2^t)\frac{\partial \gamma_2^t}{\partial \alpha_2^t}, \tag{A103}$$

$$\frac{\partial \gamma_{z,2}^t}{\partial \epsilon_1^t} = \sin(\epsilon_2^t)\frac{\partial \gamma_2^t}{\partial \epsilon_1^t}, \tag{A104}$$

$$\frac{\partial \gamma_{z,2}^t}{\partial \epsilon_2^t} = \cos(\epsilon_2^t)\gamma_2^t + \sin(\epsilon_2^t)\frac{\partial \gamma_2^t}{\partial \epsilon_2^t}, \tag{A105}$$

$$\frac{\partial \gamma_1^t}{\partial \alpha_1^t} = \cos(\epsilon_1^t)\sin(\alpha_1^t)(x_1 - x_2) - \cos(\alpha_1^t)\cos(\epsilon_1^t)(y_1 - y_2)$$
$$- \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\sin(\epsilon_2^t)(z_1 - z_2)$$
$$+ \cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\epsilon_2^t)(z_1 - z_2)$$
$$- \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\alpha_2^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)\sin(\alpha_1^t)(x_1 - x_2)$$
$$+ \cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_1^t)\cos(\alpha_2^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)(x_1 - x_2)$$
$$- \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)\sin(\alpha_1^t)\sin(\alpha_2^t)(y_1 - y_2)$$
$$+ \cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)\sin(\alpha_2^t)(y_1 - y_2), \tag{A106}$$

$$\frac{\partial \gamma_1^t}{\partial \alpha_2^t} = \sin(\epsilon_2^t)\cos(\alpha_2^t)\cos(\epsilon_2^t)\sin(\epsilon_1^t)(y_1 - y_2)$$
$$- \sin(\epsilon_2^t)\cos(\epsilon_2^t)\sin(\alpha_2^t)\sin(\epsilon_1^t)(x_1 - x_2)$$
$$+ \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\sin(\epsilon_2^t)(z_1 - z_2)$$
$$- \cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\epsilon_2^t)(z_1 - z_2)$$
$$+ \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\alpha_1^t)\cos(\alpha_2^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)(y_1 - y_2)$$
$$+ \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\alpha_2^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)\sin(\alpha_1^t)(x_1 - x_2)$$
$$-2\cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_1^t)\cos(\alpha_2^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)(x_1 - x_2)$$
$$- \cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)\sin(\alpha_2^t)(y_1 - y_2)$$
$$+2\cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_2^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)\sin(\alpha_1^t)(y_1 - y_2)$$
$$- \cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)\sin(\alpha_1^t)\sin(\alpha_2^t)(x_1 - x_2), \tag{A107}$$

$$\frac{\partial \gamma_1^t}{\partial \epsilon_1^t} = -\cos(\epsilon_1^t)(z_1 - z_2) + \cos(\alpha_1^t)\sin(\epsilon_1^t)(x_1 - x_2)$$
$$+ \sin(\alpha_1^t)\sin(\epsilon_1^t)(y_1 - y_2)$$
$$+ \sin(\epsilon_2^t)\cos(\epsilon_1^t)\sin(\epsilon_2^t)(z_1 - z_2)$$
$$+ \sin(\epsilon_2^t)\cos(\alpha_2^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)(x_1 - x_2)$$
$$+ \sin(\epsilon_2^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)\sin(\alpha_2^t)(y_1 - y_2)$$
$$- \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\alpha_1^t)\sin(\epsilon_1^t)\sin(\epsilon_2^t)(z_1 - z_2)$$
$$- \cos(\epsilon_2^t)\sin(\alpha_2^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)\sin(\epsilon_2^t)(z_1 - z_2)$$
$$- \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\alpha_1^t)\cos(\alpha_2^t)\cos(\epsilon_2^t)\sin(\epsilon_1^t)(x_1 - x_2)$$
$$- \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_2^t)\sin(\alpha_2^t)\sin(\epsilon_1^t)(y_1 - y_2)$$
$$- \cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_2^t)\cos(\epsilon_2^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)(x_1 - x_2)$$
$$- \cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\epsilon_2^t)\sin(\alpha_1^t)\sin(\alpha_2^t)\sin(\epsilon_1^t)(y_1 - y_2), \tag{A108}$$

$$\frac{\partial \gamma_1^t}{\partial \epsilon_2^t} = 2\cos(\epsilon_2^t)\sin(\epsilon_1^t)\sin(\epsilon_2^t)(z_1 - z_2)$$
$$+ \cos(\epsilon_2^t)\cos(\alpha_2^t)\cos(\epsilon_2^t)\sin(\epsilon_1^t)(x_1 - x_2)$$
$$+ \cos(\epsilon_2^t)\cos(\epsilon_2^t)\sin(\alpha_2^t)\sin(\epsilon_1^t)(y_1 - y_2)$$
$$- \sin(\epsilon_2^t)\cos(\alpha_2^t)\sin(\epsilon_1^t)\sin(\epsilon_2^t)(x_1 - x_2)$$
$$- \sin(\epsilon_2^t)\sin(\alpha_2^t)\sin(\epsilon_1^t)\sin(\epsilon_2^t)(y_1 - y_2)$$
$$- \cos(\alpha_2^t)\sin(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\epsilon_2^t)(z_1 - z_2)$$
$$+ \cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)\sin(\alpha_1^t)(z_1 - z_2)$$
$$- \sin(\alpha_2^t)\sin(\epsilon_2^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\sin(\epsilon_2^t)(z_1 - z_2)$$
$$+ \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)(z_1 - z_2)$$
$$-2\cos(\alpha_2^t)\sin(\epsilon_2^t)\cos(\alpha_1^t)\cos(\alpha_2^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)(x_1 - x_2)$$
$$-2\cos(\alpha_2^t)\sin(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)\sin(\alpha_2^t)(y_1 - y_2)$$
$$-2\sin(\alpha_2^t)\sin(\epsilon_2^t)\cos(\alpha_2^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)\sin(\alpha_1^t)(x_1 - x_2)$$
$$-2\sin(\alpha_2^t)\sin(\epsilon_2^t)\cos(\epsilon_1^t)\cos(\epsilon_2^t)\sin(\alpha_1^t)\sin(\alpha_2^t)(y_1 - y_2), \tag{A109}$$

$$\frac{\partial \gamma_2^t}{\partial \alpha_1^t} = \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\epsilon_1^t)^2(y_1 - y_2)$$
$$+ \cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\epsilon_1^t)^2(x_1 - x_2)$$
$$-2\cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\alpha_1^t)^2\cos(\epsilon_1^t)^2(y_1 - y_2)$$
$$- \sin(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\epsilon_1^t)(y_1 - y_2)$$
$$-2\cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_1^t)^2\cos(\epsilon_1^t)^2(x_1 - x_2)$$
$$+ \sin(\epsilon_2^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)(x_1 - x_2)$$
$$+ \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)(z_1 - z_2)$$
$$- \cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\epsilon_1^t)(z_1 - z_2)$$
$$+2\cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)^2\sin(\alpha_1^t)(x_1 - x_2)$$
$$-2\cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)^2\sin(\alpha_1^t)(y_1 - y_2), \tag{A110}$$

$$\frac{\partial \gamma_2^t}{\partial \alpha_2^t} = \cos(\alpha_2^t)\cos(\epsilon_2^t)(y_1 - y_2)$$
$$- \cos(\epsilon_2^t)\sin(\alpha_2^t)(x_1 - x_2)$$
$$+ \cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_1^t)^2\cos(\epsilon_1^t)^2(x_1 - x_2)$$
$$- \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\epsilon_1^t)^2\sin(\alpha_1^t)^2(y_1 - y_2)$$
$$- \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)(z_1 - z_2)$$
$$+ \cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\epsilon_1^t)(z_1 - z_2)$$
$$- \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)^2\sin(\alpha_1^t)(x_1 - x_2)$$
$$+ \cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)^2\sin(\alpha_1^t)(y_1 - y_2), \tag{A111}$$

$$\frac{\partial \gamma_2^t}{\partial \epsilon_1^t} = -2\sin(\epsilon_2^t)\cos(\epsilon_1^t)\sin(\epsilon_1^t)(z_1 - z_2)$$
$$- \sin(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)^2(x_1 - x_2)$$
$$+ \sin(\epsilon_2^t)\cos(\alpha_1^t)\sin(\epsilon_1^t)^2(x_1 - x_2)$$
$$- \sin(\epsilon_2^t)\cos(\epsilon_1^t)^2\sin(\alpha_1^t)(y_1 - y_2)$$
$$+ \sin(\epsilon_2^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)^2(y_1 - y_2)$$
$$- \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)^2(z_1 - z_2)$$
$$+ \cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\alpha_1^t)\sin(\epsilon_1^t)^2(z_1 - z_2)$$
$$- \cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\epsilon_1^t)^2\sin(\alpha_1^t)(z_1 - z_2)$$
$$+ \cos(\epsilon_2^t)\sin(\alpha_2^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)^2(z_1 - z_2)$$
$$+2\cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\alpha_1^t)^2\cos(\epsilon_1^t)\sin(\epsilon_1^t)(x_1 - x_2)$$
$$+2\cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)^2\sin(\epsilon_1^t)(y_1 - y_2)$$
$$+2\cos(\alpha_2^t)\cos(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)(y_1 - y_2)$$
$$+2\cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)(x_1 - x_2), \tag{A112}$$

$$\begin{aligned}\frac{\partial \gamma_2^t}{\partial \epsilon_2^t} &= \cos(\epsilon_2^t)(z_1 - z_2)\\ &\quad - \cos(\epsilon_2^t)\sin(\epsilon_1^t)^2(z_1 - z_2)\\ &\quad - \cos(\alpha_2^t)\sin(\epsilon_2^t)(x_1 - x_2)\\ &\quad - \sin(\alpha_2^t)\sin(\epsilon_2^t)(y_1 - y_2)\\ &\quad + \sin(\alpha_2^t)\sin(\epsilon_2^t)\cos(\epsilon_1^t)^2\sin(\alpha_1^t)^2(y_1 - y_2)\\ &\quad - \cos(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\epsilon_1^t)(x_1 - x_2)\\ &\quad - \cos(\epsilon_2^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)(y_1 - y_2)\\ &\quad + \cos(\alpha_2^t)\sin(\epsilon_2^t)\cos(\alpha_1^t)^2\cos(\epsilon_1^t)^2(x_1 - x_2)\\ &\quad + \cos(\alpha_2^t)\sin(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\epsilon_1^t)(z_1 - z_2)\\ &\quad + \sin(\alpha_2^t)\sin(\epsilon_2^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)(z_1 - z_2)\\ &\quad + \cos(\alpha_2^t)\sin(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)^2\sin(\alpha_1^t)(y_1 - y_2)\\ &\quad + \sin(\alpha_2^t)\sin(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)^2\sin(\alpha_1^t)(x_1 - x_2),\end{aligned} \tag{A113}$$

$$\begin{aligned}\frac{\partial \kappa_{12}^t}{\partial \alpha_1^t} &= 2\cos(\epsilon_2^t)\cos(\epsilon_1^t)(\cos(\alpha_2^t)\sin(\epsilon_2^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)\\ &\quad - \sin(\alpha_2^t)\sin(\epsilon_2^t)\cos(\alpha_1^t)\sin(\epsilon_1^t)\\ &\quad + \cos(\alpha_2^t)\cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\epsilon_1^t)\\ &\quad - \cos(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\\ &\quad + 2\cos(\alpha_2^t)^2\cos(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\\ &\quad - 2\cos(\alpha_2^t)\cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_1^t)^2\cos(\epsilon_1^t)),\end{aligned} \tag{A114}$$

$$\begin{aligned}\frac{\partial \kappa_{12}^t}{\partial \alpha_2^t} &= -2\cos(\epsilon_2^t)\cos(\epsilon_1^t)\left(\cos(\alpha_2^t)\sin(\epsilon_2^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)\right.\\ &\quad - \sin(\alpha_2^t)\sin(\epsilon_2^t)\cos(\alpha_1^t)\sin(\epsilon_1^t)\\ &\quad + \cos(\alpha_2^t)\cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\epsilon_1^t)\\ &\quad - \cos(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\\ &\quad + 2\cos(\alpha_2^t)^2\cos(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\\ &\quad \left. - 2\cos(\alpha_2^t)\cos(\epsilon_2^t)\sin(\alpha_2^t)\cos(\alpha_1^t)^2\cos(\epsilon_1^t)\right),\end{aligned} \tag{A115}$$

$$\begin{aligned}\frac{\partial \kappa_{12}^t}{\partial \epsilon_1^t} &= 2\cos(\epsilon_2^t)\sin(\alpha_2^t)\sin(\epsilon_2^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)^2\\ &\quad - 2\sin(\epsilon_2^t)^2\cos(\epsilon_1^t)\sin(\epsilon_1^t)\\ &\quad + 2\cos(\alpha_2^t)^2\cos(\epsilon_2^t)^2\cos(\alpha_1^t)^2\cos(\epsilon_1^t)\sin(\epsilon_1^t)\\ &\quad + 2\cos(\epsilon_2^t)^2\sin(\alpha_2^t)^2\cos(\epsilon_1^t)\sin(\alpha_1^t)^2\sin(\epsilon_1^t)\\ &\quad - 2\cos(\alpha_2^t)\cos(\epsilon_2^t)\sin(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)^2\\ &\quad + 2\cos(\alpha_2^t)\cos(\epsilon_2^t)\sin(\epsilon_2^t)\cos(\alpha_1^t)\sin(\epsilon_1^t)^2\\ &\quad - 2\cos(\epsilon_2^t)\sin(\alpha_2^t)\sin(\epsilon_2^t)\cos(\epsilon_1^t)^2\sin(\alpha_1^t)\\ &\quad + 4\cos(\alpha_2^t)\cos(\epsilon_2^t)^2\sin(\alpha_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\sin(\epsilon_1^t),\end{aligned} \tag{A116}$$

$$\begin{aligned}\frac{\partial \kappa_{12}^t}{\partial \epsilon_2^t} &= 2\cos(\alpha_2^t)\sin(\epsilon_2^t)^2\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\epsilon_1^t)\\ &\quad - 2\cos(\alpha_2^t)\cos(\epsilon_2^t)^2\cos(\alpha_1^t)\cos(\epsilon_1^t)\sin(\epsilon_1^t)\\ &\quad - 2\cos(\epsilon_2^t)\sin(\epsilon_2^t)\sin(\epsilon_1^t)^2\\ &\quad - 2\cos(\epsilon_2^t)^2\sin(\alpha_2^t)\cos(\epsilon_1^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)\\ &\quad + 2\cos(\alpha_2^t)^2\cos(\epsilon_2^t)\sin(\epsilon_2^t)\cos(\alpha_1^t)^2\cos(\epsilon_1^t)^2\\ &\quad + 2\sin(\alpha_2^t)\sin(\epsilon_2^t)^2\cos(\epsilon_1^t)\sin(\alpha_1^t)\sin(\epsilon_1^t)\\ &\quad + 2\cos(\epsilon_2^t)\sin(\alpha_2^t)^2\sin(\epsilon_2^t)\cos(\epsilon_1^t)^2\sin(\alpha_1^t)^2\\ &\quad + 4\cos(\alpha_2^t)\cos(\epsilon_2^t)\sin(\alpha_2^t)\sin(\epsilon_2^t)\cos(\alpha_1^t)\cos(\epsilon_1^t)^2\sin(\alpha_1^t).\end{aligned} \tag{A117}$$

REFERENCES

[1] J. Arvo
Graphics Gems II. New York, NY: Academic Press, 1991.

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

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

[4] D. Belfadel, Y. Bar-Shalom, and P. Willett
"Simultaneous target state and passive sensors bias estimation,"
in *Proc. 19th Int. Conf. Inf. Fusion*, Jul. 2016, pp. 1223–1227.

[5] D. Belfadel, R. W. Osborne, and Y. Bar-Shalom
"Bias estimation for moving optical sensor measurements with targets of opportunity,"
Proc. SPIE, vol. 9092, no. 11, pp. 1805–1812, 2014.

[6] D. Belfadel, R. Osborne, and Y. Bar-Shalom
"Bias estimation and observability for optical sensor measurements with targets of opportunity,"
J. Adv. Inf. Fusion, vol. 9, no. 2, pp. 59–74, Dec. 2014.

[7] D. Belfadel, R. W. Osborne, and Y. Bar-Shalom
"Bias estimation for optical sensor measurements with targets of opportunity,"
in *Proc. 16th Int. Conf. Inf. Fusion*, Jul. 2013, pp. 1805–1812.

[8] M. F. Bugallo, T. Lu, and P. M. Djuric
"Bearings-only tracking with biased measurements,"
in *Proc. 2nd IEEE Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process.*, 2007, pp. 265–268.

[9] H. Chen and F. Han
"Bias estimation for multiple passive sensors,"
in *Proc. 2012 Int. Conf. Meas., Inf. Control (MIC)*, vol. 2, May 2012, pp. 1081–1084.

[10] T. M. Clemons and K. C. Chang
"Bias correction using background stars for space-based IR tracking,"
in *Proc. 12th Int. Conf. Inf. Fusion*, Jul. 2009, pp. 2028–2035.

[11] T. M. Clemons and K. C. Chang
"Sensor calibration using in-situ celestial observations to estimate bias in space-based missile tracking,"
IEEE Trans. Aerosp. Electron. Syst., vol. 48, no. 2, pp. 1403–1427, Apr. 2012.

[12] D. Crouse, Y. Bar-Shalom, and P. Willett
"Sensor bias estimation in the presence of data association uncertainty,"
Proc. SPIE, vol. 7445, no. 28, p. 74450P, Aug. 2009.

[13] S. Fortunati, A. Farina, F. Gini, A. Graziano, M. S. Greco, and S. Giompapa
"Least squares estimation and Cramér–Rao type lower bounds for relative sensor registration process,"
IEEE Trans. Signal Process., vol. 59, no. 3, pp. 1075–1087, Mar. 2011.

[14] S. Fortunati, F. Gini, M. Greco, A. Farina, A. Graziano, and S. Giompapa
"An identifiability criterion in the presence of random nuisance parameters,"

in *Proc. 20th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2012, pp. 1194–1198.

[15] R. Goldman
Graphics Gems. New York, NY: Academic Press, 1990.

[16] Y. Hu and D. Zhou
"Bias fusion estimation for multi-target tracking systems with multiple asynchronous sensors,"
Aerosp. Sci. Technol., vol. 27, no. 1, pp. 95–104, 2013.

[17] M. Kowalski, Y. Bar-Shalom, P. Willett, and T. Fair
"Unbiased conversion of passive sensor measurements,"
in *Proc. IEEE 8th Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process. (CAMSAP)*, Dec. 2019, pp. 505–509.

[18] M. Kowalski, Y. Bar-Shalom, P. Willett, and T. Fair
"Unbiased conversion of passive sensor measurements using closest point of approach,"
IEEE Trans. Aerosp. Electron. Syst., to be published.

[19] M. Kowalski, Y. Bar-Shalom, P. Willett, B. Milgrom, and R. Ben-Dov
"CRLB for multi-sensor rotational bias estimation for passive sensors without target state estimation,"
in *Proc. SPIE Signal Process., Sensor/Inf. Fusion, Target Recognit. XXVIII*, vol. 11018, Apr. 2019, pp. 33–41.

[20] X. Lin and Y. Bar-Shalom
"Multisensor target tracking performance with bias compensation,"
IEEE Trans. Aerosp. Electron. Syst., vol. 42, no. 3, pp. 1139–1149, Jul. 2006.

[21] X. Lin, Y. Bar-Shalom, and T. Kirubarajan
"Multisensor multitarget bias estimation for general asynchronous sensors,"
IEEE Trans. Aerosp. Electron. Syst., vol. 41, no. 3, pp. 899–921, Jul. 2005.

[22] X. Lin, Y. Bar-Shalom, and T. Kirubarajan
"Exact multisensor dynamic bias estimation with local tracks,"
IEEE Trans. Aerosp. Electron. Syst., vol. 40, no. 2, pp. 576–590, Apr. 2004.

[23] M. Longbin, S. Xiaoquan, Z. Yiyu, S. Z. Kang, and Y. Bar-Shalom
"Unbiased converted measurements for tracking,"
IEEE Trans. Aerosp. Electron. Syst., vol. 34, no. 3, pp. 1023–1027, Jul. 1998.

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

[25] Q. Song and Y. He
"A real-time registration algorithm for passive sensors with TOA and angle biases,"
in *Proc. 3rd Int. Congr. Image Signal Process.*, vol. 9, 2010, pp. 4170–4173.

[26] E. Taghavi, R. Tharmarasa, T. Kirubarajan, and M. Mcdonald
"Multisensor–multitarget bearing-only sensor registration,"
IEEE Trans. Aerosp. Electron. Syst., vol. 52, no. 4, pp. 1654–1666, Aug. 2016.

[27] B.-L. Xu and Z. Wang
"Biased bearings-only parameter estimation for bistatic system,"
J. Electron. (China), vol. 24, pp. 326–331, May 2007.

[28] W. Xu
"Cramér–Rao bound for bearing estimation with bias correction,"
in *Proc. OCEANS 2007*, 2007.

Michael Kowalski received the B.S. degree in electrical engineering from the University of Connecticut, Storrs, CT, USA, in 2015, and is currently working toward the Ph.D. degree in electrical engineering at the Department of Electrical and Computer Engineering, University of Connecticut. His research focuses on bias estimation as well as sensor fusion, and target tracking.

Yaakov Bar-Shalom (F'84) received the B.S. and M.S. degrees in electrical engineering from the Technion, Haifa, Israel, in 1963 and 1967, respectively, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, USA, in 1970. He is currently a Board of Trustees Distinguished Professor in the Department of Electrical and Computer Engineering and Marianne E. Klewin Professor at the University of Connecticut, Storrs, CT, USA. His current research interests are in estimation theory, target tracking, and data fusion. He has authored 600 papers and book chapters. He has coauthored/edited eight books, including *Tracking and Data Fusion* (YBS Publishing, 2011). He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He was an Associate Editor of the IEEE Transactions on Automatic Control and *Automatica*. He was General Chairman of the 1985 ACC, General Chairman of FUSION 2000, President of ISIF in 2000 and 2002, and Vice President for Publications during 2004–2013. Since 1995, he has been a Distinguished Lecturer of the IEEE AESS. He was the corecipient of the M. Barry Carlton Award for the best paper in the IEEE TAESystems in 1995 and 2000. In 2002, he was the recipient of 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 was the recipient of 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 was the recipient of the 2015 ISIF Award for a Lifetime of Excellence in Information Fusion. This award has been renamed in 2016 as the Yaakov Bar-Shalom Award for a Lifetime of Excellence in Information Fusion. He has the following Wikipedia page: https://en.wikipedia.org/wiki/Yaakov Bar-Shalom.

Peter Willett received the B.A.S. degree in engineering science from the University of Toronto, 1982, and the M.E., M.Sc., and Ph.D. degrees in electrical engineering from the Princeton University, Princeton, NJ, USA, in 1983, 1984, and 1986, respectively. He has been a faculty member with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA, since 1986. Since 1998 he has been a Professor, and since 2003 an IEEE Fellow. His primary areas of research have been statistical signal processing, detection, machine learning, communications, data fusion, and tracking. He is Chief Editor for the IEEE AESS Magazine (2018–2020). He was Editor-in-Chief for the IEEE Signal Processing Letters, 2014–2016, and before that for the IEEE Transactions on Aerospace and Electronic Systems from 2006 to 2011. He was also AESS Vice President for Publications from 2012 to 2014. He is a member of the IEEE Fellows Committee, Ethics Committee, and Periodicals Committee, as well as the IEEE Signal Processing Society's Technical Activities and Conference Boards. He is Chief Editor for the IEEE AES Magazine. He is a member of the IEEE AESS Board of Governors and was Chair of the IEEE Signal Processing Society's Sensor-Array and Multichannel (SAM) technical committee.

Bias Estimation for Collocated Sensors: Model Identification and Measurement Fusion

KAIPEI YANG
YAAKOV BAR-SHALOM
PETER WILLETT
HIROSHI INOU

The sensor bias estimation problem is crucial in autonomous driving systems for perception and target tracking. This work considers the bias estimation for two collocated synchronized sensors with slowly varying additive biases. The differences between the two sensors' observations are used to eliminate the target state. Consequently, the bias estimation is independent of the target-state estimation. The biases' observability condition is met when the two sensors' biases are Ornstein–Uhlenbeck stochastic processes with different time constants. The bias models, including the time constants and measurement noises, can be identified based on a sample autocorrelation or using the maximum-likelihood estimation technique. A maximum-likelihood measurement fusion technique is introduced for the bias-compensated observations. Simulation results, for several scenarios with various bias model parameters, prove the consistency of the estimator. It is shown that the uncertainties of biases are significantly reduced by the estimation algorithm presented. The sensitivity of the proposed algorithm is also tested with mismatched filters as well as the estimated bias models. Finally, the benefits of bias estimation in measurement fusion are evaluated.

Manuscript received May 23, 2020; revised May 27, 2020; released for publication September 12, 2020; corrected November 2, 2021.

Associate Editor: Ramona Georgescu
Refereeing of this contribution was handled by Florian Meyer.
K. Yang, Y. Bar-Shalom, and P. Willett are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269, USA (E-mail: kaipei.yang, yaakov.bar-shalom, peter.willett@uconn.edu). H. Inou is with the DENSO International America, Inc., Southfield, MI 48033, USA (E-mail: hiroshi.inou.j5f@jp.denso.com).

1557-6418/20/$17.00

I. INTRODUCTION

Target tracking has always been an important problem for autonomous driving systems where multiple sensors are utilized to improve the tracking accuracy. Unfortunately, these sensors, such as radars, lidars, and cameras, are prone to biases, which can lead to a problematic association and hence poor results in target tracking. The sensors used in autonomous driving vehicles can only be placed together or very close (practically collocated), which makes the bias estimation challenging. Consequently, only a few works partially addressed this problem. Sensor calibration via off-line preprocessing is supposed to eliminate the sensor biases. However, it requires the knowledge of ground truth, and to be of value, the biases must be time-invariant. For the case where the sensor biases are dynamic and slowly varying, off-line calibration is not sufficient. This work proposes a real-time bias-estimation method for collocated sensors independent of the target-motion tracking and approaches for identifying the bias models.

Kastella and Yeary [7] considered the bias-estimation problem for radars on moving platforms by decoupling the tracking of targets of opportunity and the estimation of the radar and platform biases. Lin et al. [10] solved exact bias estimation for an active sensor by using pairs of range and angle measurements to create pseudo-measurements of the biases of both sensors relying on the nonlinearity of the range and angle measurements. Bar-Shalom [3] considered time-varying bias estimation along with the target state. This work is based on [3] and uses the subtraction between the sensor measurements as in [10] to eliminate the target state in estimating the sensor biases, i.e., the biases can be estimated independently of the target. In [4], the authors considered the problem of estimating sensor biases from measurements of targets flying on known trajectories by augmenting the kinematic state vector with sensor bias parameters. In [10] and [11], the bias model includes scale biases and unknown locations of the sensors, as well as the usual offset biases. Kowalski et al. [8] considered three-dimensional sensor bias estimation using sine space measurements and showed the achievability of the Cramér–Rao lower bound.

To handle bias estimation for collocated sensors, this work considers slowly varying sensor biases that are modeled as Ornstein–Uhlenbeck (OU, a class of Gauss–Markov) processes (as discussed in [3]) and deals with bias estimation with the following contributions: (i) solving the problem for collocated synchronized sensors; (ii) estimation of biases independent of target motion by using the difference between the associated sensor observations; and (iii) application of the proposed method to all kinds of observations (i.e., bearings, range, etc.) from various types of sensors.

The bias models (parameters of the OU process) are typically unknown with limited prior information. In [12] and [13], the mean-reverting OU process parameter

estimation is discussed along with a long-term prediction. Two approaches are introduced in this paper to identify the bias model: (i) sample autocorrelation based method; and (ii) maximum-likelihood (ML) estimation of the model parameters. The prior information about the sensor biases consists of initial distributions, assumed to be Gaussian with zero mean and certain variances. After the bias estimation, one can fuse the local observations, according to Fusion Configuration III [2], with the bias compensation taking into account the error in the bias estimates. The fusion is carried out using the ML criterion. The performance of the proposed method is tested via simulations based on Monte Carlo (MC) runs by showing the reduction in the mean-square (MS) error of the bias-compensated fused measurements versus fusion without compensation. The estimator and fuser presented are shown to be consistent.

The process flowchart for bias estimation, fusion, and system identification is shown in Fig. 1. This paper is organized as follows. Section II formulates the problem by introducing the bias dynamic model and the bias measurement model (using the subtraction between the sensor observations) and discusses the bias observability. The bias model identification methods are presented in Section III. In Section IV, the fusion of the bias-compensated observations is presented. Section V gives the simulation results of several scenarios. Conclusions and remarks are in Section VI.

II. PROBLEM FORMULATION

The challenge of this work is to estimate the (collocated) sensor biases efficiently given synchronized observations defined as

$$z_1(k) = h\left[\mathbf{x}(k), \mathbf{s}(k)\right] + b_1(k) + w_1(k), \qquad k = 1, 2, \ldots, N, \tag{1}$$

and

$$z_2(k) = h\left[\mathbf{x}(k), \mathbf{s}(k)\right] + b_2(k) + w_2(k), \qquad k = 1, 2, \ldots, N, \tag{2}$$

where $\mathbf{x}$ is the true (common) target state, which is unknown, $\mathbf{s}$ is the sensor state, and $h[\cdot, \cdot]$ is the generic observation model (angle or range). Since the sensors are collocated, they share the same sensor motion. The observations obtained from the sensors depend on both the sensor and target motions as well as the biases and noises. The bias estimation introduced in the following does not require the target state. The observation noises

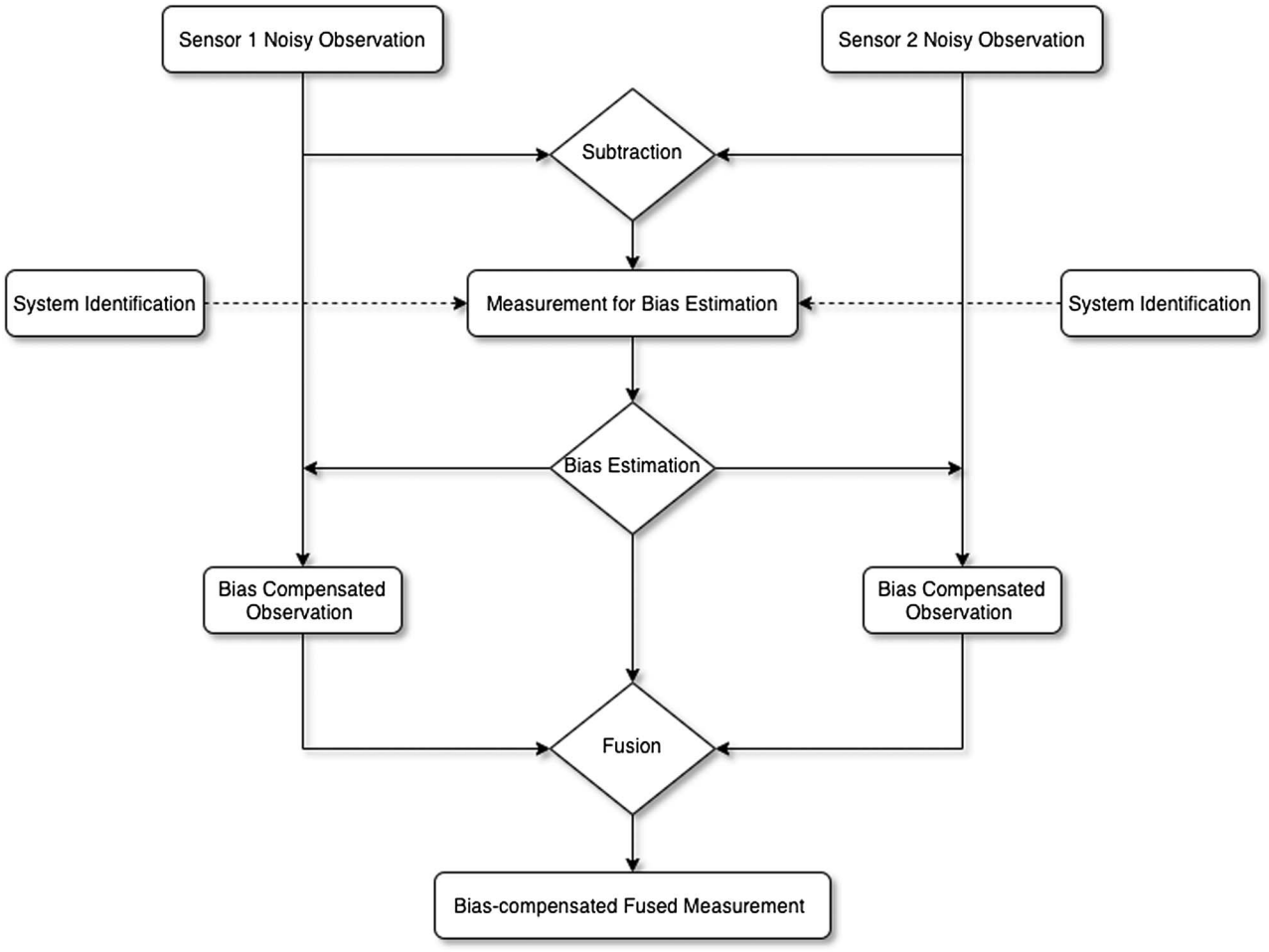


Fig. 1. Process flowchart for bias estimation and fusion.

w_1 and w_2 are assumed to be zero mean, white with variances $\sigma^2_{w_1}$ and $\sigma^2_{w_2}$, and are independent of each other and of the sensor biases. The above observations are not restricted to the type of sensors; only the same kind of observation is required, i.e., radars, lidars, and cameras can provide angle observations; lidars and radars can also provide range observations. It will be shown in the sequel that bias estimation is independent of the target-state estimation, i.e., bias estimation and state tracking are decoupled. The observations considered are using generic models in one coordinate (i.e., with a dimension of 1) to aid in the clarity of the exposition.

The sensor biases are slowly varying, modeled as an OU (first-order Gauss–Markov) process [7]. The discrete dynamic model used for the biases [1], [3] is (two collocated sensors are considered in this work)

$$b_i(k+1) = \alpha_i b_i(k) + v_i(k), \qquad i = 1, 2, \tag{3}$$

with

$$\alpha_i = e^{-T/\tau_i}, \tag{4}$$

where T is the sampling interval and τ_i is the time constant of the bias evolution (assumed to be known; its estimation is discussed in Section III).

The time constant is given in terms of α_i as

$$\tau_i = -T \ln \alpha_i. \tag{5}$$

The above expression can be rewritten using the first-order Taylor expansion for $\tau \gg T$ as

$$\tau_i \approx \frac{T}{1-\alpha_i}. \tag{6}$$

The driving process noises v_i are assumed to be zero mean, white with variances $\sigma^2_{v_i}$. All the noises v_i and w_i are independent. Using the above model guarantees that the bias estimates are bounded since (3) is stable.

The MS value of the bias b_i is $\sigma^2_{b_i}$ and its relationship to the corresponding process noise variance is

$$\sigma^2_{v_i} = (1-\alpha_i^2)\sigma^2_{b_i}. \tag{7}$$

Since the sensor biases b_i will be estimated independently of the target state, only the difference between the observations is used:

$$z(k) = z_1(k) - z_2(k) = b_1(k) - b_2(k) + w_1(k) - w_2(k). \tag{8}$$

The bias state to be estimated is

$$\mathbf{b}(k) = [b_1(k)\ b_2(k)]' \tag{9}$$

with the state equation

$$\mathbf{b}(k+1) = F\mathbf{b}(k) + \mathbf{v}(k), \tag{10}$$

where

$$F = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix}, \tag{11}$$

and the process noise vector is

$$\mathbf{v} = [v_1\ v_2]' \tag{12}$$

with the covariance matrix

$$Q = \begin{bmatrix} \sigma^2_{v_1} & 0 \\ 0 & \sigma^2_{v_2} \end{bmatrix}. \tag{13}$$

The measurement model based on (8) is

$$z(k) = H\mathbf{b}(k) + w(k), \tag{14}$$

where

$$H = \begin{bmatrix} 1 & -1 \end{bmatrix} \tag{15}$$

and the measurement noise $w(k)$ is

$$w(k) = w_1(k) - w_2(k), \tag{16}$$

which has variance $\sigma^2_{w_1} + \sigma^2_{w_2}$.

A Kalman filter (KF) is then used for the estimation of $\mathbf{b}(k)$, which gives the estimate at time k [1]

$$\hat{\mathbf{b}}(k) = [\hat{b}_1(k)\ \hat{b}_2(k)]'. \tag{17}$$

The observability of the above system can be verified via the observability matrix [9]

$$\mathcal{O} = \begin{bmatrix} H \\ HF \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ \alpha_1 & -\alpha_2 \end{bmatrix}, \tag{18}$$

which has full rank under the conditions (i) $\alpha_1 \neq \alpha_2$, and (ii) none is unity.[1] Assuming that the sensors are different, the bias models will be different due to their physical properties, i.e., different α_i in (3). Using a discrete-time Wiener process to model both sensor biases will lead to lack of observability of the system with $\alpha_i = 1$, and the bias becomes the integral (sum) of the white noise sequence terms (and diverges). Thus, both the observability conditions can be met with reasonably realistic assumptions.

It can be easily seen that the pair (F, C), where C is the Cholesky factor of the covariance matrix Q, is completely controllable. Thus, the solution of the discrete-time Riccati equation for such a time-invariant system will converge to a finite steady-state (SS) covariance [1], which can obtained via KF.

III. BIAS MODEL IDENTIFICATION

The sensor biases are assumed to be OU processes, as shown in (3); however, the time constants τ_i are, in general, unknown. To identify the system (bias model) as well as the process and measurement noise variances, two approaches are introduced: (i) system parameter estimation based on sample autocorrelation, and (ii) ML estimation. Note that the bias model identification is independent of and prior to the bias estimation.

For simplicity, only one system is analyzed for illustration; however, the approach can be used for multiple sensors with the same model but different parameters.

[1] The bias model should not diverge.

Consider the discrete-time OU bias model (with a slowly varying bias[2])

$$b(k+1) = \alpha b(k) + v(k) \tag{19}$$

and noisy observation model ($k = 1, 2, \ldots, N$)

$$z = h[\mathbf{x}(k), \mathbf{s}(k)] + b(k) + w(k), \tag{20}$$

$$o(k) \triangleq z(k) - h[\mathbf{x}(k), \mathbf{s}(k)] = b(k) + w(k), \tag{21}$$

with α defined in (4), v is the process noise with variance σ_v^2, and w is the measurement noise with variance σ_w^2; $\mathbf{x}$ and $\mathbf{s}$ are as defined in (2). The observation $o(k)$ is obtained assuming that the truth is known, which can be done in the off-line precalibration. However, note that $h[\mathbf{x}(k), \mathbf{s}(k)]$ will cancel in (8), which will be used in estimating the two biases (17). The identification of the bias models will rely on (21).

A. Bias Model Parameter Estimation Using Sample Autocorrelation

The autocorrelation of $o(k)$, assumed to be wide-sense stationary (WSS), is

$$\begin{aligned} r(m) &= \mathrm{E}\{o(k)o(k-m)\} \\ &= \mathrm{E}\{[b(k) + w(k)][b(k-m) + w(k-m)]\} \\ &= \mathrm{E}\{b(k)b(k-m)\} + \sigma_w^2 \delta(m) \\ &= r_b(m) + \sigma_w^2 \delta(m), \end{aligned} \tag{22}$$

where $\delta(k)$ is the Kronecker delta function and $r_b(m)$ is the autocorrelation of $b(k)$, also assumed to be WSS,

$$\begin{aligned} r_b(m) &= \mathrm{E}\{b(k)b(k-m)\} \\ &= \mathrm{E}\{[\alpha b(k-1) + v(k-1)][b(k-m)]\} \\ &= \alpha^m r_b(0) \end{aligned} \tag{23}$$

and

$$\begin{aligned} r_b(0) &= \mathrm{E}\{b(k)b(k)\} \\ &= \mathrm{E}\{[\alpha b(k-1) + v(k-1)] \\ &\quad \cdot [\alpha b(k-1) + v(k-1)]^*\} \\ &= \alpha^2 r_b(0) + \sigma_v^2. \end{aligned} \tag{24}$$

The above equation yields

$$r_b(0) = \frac{\sigma_v^2}{1-\alpha^2}. \tag{25}$$

Substituting (25) into (22) gives

$$r(m) = \alpha^m \frac{\sigma_v^2}{1-\alpha^2} + \sigma_w^2 \delta(m), \tag{26}$$

which can be used to estimate α, σ_v^2, and σ_w^2. That is, assuming that the sample autocorrelations are available (sufficiently accurate since one can have only sample autocorrelations, i.e., time averages), one has

$$\alpha = \frac{r(2)}{r(1)}, \tag{27}$$

$$\sigma_v^2 = \frac{r(1)^2 - r(2)^2}{r(2)}, \tag{28}$$

$$\sigma_w^2 = r(0) - \frac{r(1)^2}{r(2)}. \tag{29}$$

Note that the above solution is highly dependent on the accuracy of sample autocorrelations (especially when α is very close to 1), which cannot be guaranteed with limited sample data. Assuming that more autocorrelations (i.e., more than $r(2)$) are available, the parameters can be estimated using all sample autocorrelations to improve the accuracy as discussed in the following.

Equation (26) can be written for $m > 0$ as

$$\ln r(m) = m\beta + \gamma, \qquad m > 0, \tag{30}$$

where

$$\beta = \ln \alpha, \tag{31}$$

$$\gamma = \ln \frac{\sigma_v^2}{1-\alpha^2}. \tag{32}$$

The least-squares (LS) estimate of parameters $\hat{\beta}$ and $\hat{\gamma}$, given $(r(1), \ldots, r(M))$, can be obtained as (see Appendix)

$$\begin{bmatrix} \hat{\beta} \\ \hat{\gamma} \end{bmatrix} = \begin{bmatrix} \sum_{m=1}^M m^2 & \sum_{m=1}^M m \\ \sum_{m=1}^M m & \sum_{m=1}^M 1 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{m=1}^M m \ln r(m) \\ \sum_{m=1}^M \ln r(m) \end{bmatrix}, \tag{33}$$

and, using (31) and (32), one has

$$\hat{\alpha} = e^{\hat{\beta}}, \tag{34}$$

$$\hat{\sigma}_v^2 = (1 - e^{2\hat{\beta}}) e^{\hat{\gamma}}. \tag{35}$$

The full set of model parameter estimates is given by (34), (35), and (29). Note that the parameter σ_w^2 does not require an additional LS estimator since it is not sensitive to the accuracy of sample autocorrelations since $\sigma_v^2 \ll \sigma_w^2$.

B. ML Estimation of the System Parameters

The above equations can provide an explicit solution for system parameter estimation in terms of the sample autocorrelations, and due to computational and speed demands, this is of high interest. However, the accuracy of these estimated autocorrelations is highly dependent on the data batch length. Alternatively, the ML approach can be used for estimating the system parameters of interest [5], [6].

[2]This implies that α is near unity and σ_v is small.

The observation model (21) can be modified taking into account the bias model (19) as

$$\begin{aligned} o(k) &= \alpha o(k-1) + w(k) - \alpha w(k-1) + v(k-1) \\ &= \alpha o(k-1) + u(k), \end{aligned} \tag{36}$$

where

$$u(k) \triangleq w(k) - \alpha w(k-1) + v(k-1) \tag{37}$$

is zero mean but not white.

Consider an observation batch of length L ($L \leq k-1$)

$$\begin{aligned} \mathbf{O}^L(k) &= \begin{bmatrix} o(k) \\ o(k-1) \\ \cdots \\ o(k-L+1) \end{bmatrix} \\ &= \alpha \begin{bmatrix} o(k-1) \\ o(k-2) \\ \cdots \\ o(k-L) \end{bmatrix} + \begin{bmatrix} u(k) \\ u(k-1) \\ \cdots \\ u(k-L+1) \end{bmatrix} \\ &= \alpha \mathbf{O}^L(k-1) + \mathbf{U}^L(k). \end{aligned} \tag{38}$$

The noise vector $\mathbf{U}^L(k)$ is zero mean with a covariance matrix of dimension L

$$R^U(L) \approx \begin{bmatrix} 2\sigma_w^2 & -\alpha\sigma_w^2 & & 0 \\ -\alpha\sigma_w^2 & \ddots & \ddots & \\ & \ddots & \ddots & -\alpha\sigma_w^2 \\ 0 & & -\alpha\sigma_w^2 & 2\sigma_w^2 \end{bmatrix} \tag{39}$$

under the assumptions that (i) α is very close to 1, and (ii) the effect of the process noise v is negligible [$\sigma_v^2 \ll \sigma_w^2$, which can be seen from (7) as the difference on the right-hand side is close to zero] and will not be estimated.[3]

The likelihood function (LF) of α and σ_w^2 based on (38) is

$$\begin{aligned} &\Lambda(\alpha, \sigma_w^2 | \mathbf{O}^L(k)) \\ &= |2\pi R^U(L)|^{-1/2} \exp\left[\frac{1}{2}\Delta_{\mathbf{O}}(k)'[R^U(L)]^{-1}\Delta_{\mathbf{O}}(k)\right], \end{aligned} \tag{40}$$

where

$$\Delta_{\mathbf{O}}(k) = \mathbf{O}^L(k) - \alpha \mathbf{O}^L(k-1). \tag{41}$$

The ML estimates (MLEs), $\hat{\alpha}$, and $\hat{\sigma}_w^2$ can be found by maximizing (40) via a numerical search. In practice, a two-dimensional grid can be used for this. The process noise σ_v^2 estimate can be obtained via (7) using

$$\hat{\sigma}_v^2 = \sigma_o^2 - \hat{\sigma}_w^2, \tag{42}$$

where σ_o^2 is the MS value of the noisy observation (21).

[3]The process noise variance σ_v^2 would appear added to each diagonal term.

IV. FUSION OF THE OBSERVATION WITH BIAS COMPENSATION

Under the Gaussian assumption, the fusion of bias-compensated observations can be solved using the ML criterion, i.e., by maximizing the LF or by minimizing the negative log-likelihood function (NLLF) of the target position based on the observations from the two sensors.[4] Note that for the linear Gaussian case, the LS estimator and ML estimator, as well as the minimum MS error (MMSE) estimator, coincide [1], [2].

With the bias estimates, the current observation can be expressed as

$$z_i(k) = \zeta(k) + \hat{b}_i(k) + \tilde{b}_i(k) + w_i(k), \qquad i = 1, 2, \tag{43}$$

where

$$\zeta(k) = h[\mathbf{x}(k), \mathbf{s}(k)] \tag{44}$$

is the noiseless fused observation that needs to be estimated given $z_i(k)$ and the bias estimates $\hat{b}_i(k)$, and by accounting for the residual bias error $\tilde{b}_i(k)$. The estimates for sensor biases are obtained using a KF and then we directly estimate the fused observation with bias compensation.

The bias-compensated ("bc") observations, omitting the time argument k for simplicity, are

$$z_1^{\text{bc}} = z_1 - \hat{b}_1 = \zeta + \tilde{b}_1 + w_1 \tag{45}$$

and

$$z_2^{\text{bc}} = z_2 - \hat{b}_2 = \zeta + \tilde{b}_2 + w_2. \tag{46}$$

Under the ML criterion, the fusion is carried out by estimating ζ based on the bias-compensated observation vector $[z_1^{\text{bc}}\ z_2^{\text{bc}}]'$. The fused observation[5] with bias compensation ("Fbc") is [1, Eq. (3.4.1-9)]

$$\begin{aligned} \hat{\zeta}^{\text{Fbc}} = &\left[(H^{\text{Fbc}})'(R^{\text{Fbc}})^{-1}H^{\text{Fbc}}\right]^{-1} \\ &\cdot (H^{\text{Fbc}})'(R^{\text{Fbc}})^{-1}[z_1^{\text{bc}}\ z_2^{\text{bc}}]', \end{aligned} \tag{47}$$

where, in this case,

$$H^{\text{Fbc}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{48}$$

and

$$R^{\text{Fbc}} = E\left\{\begin{bmatrix} \tilde{b}_1 + w_1 \\ \tilde{b}_2 + w_2 \end{bmatrix}\begin{bmatrix} \tilde{b}_1 + w_1 & \tilde{b}_2 + w_2 \end{bmatrix}\right\} \tag{49}$$

$$= \begin{bmatrix} P_{11} + \sigma_{w_1}^2 & P_{12} \\ P_{12} & P_{22} + \sigma_{w_2}^2 \end{bmatrix}. \tag{50}$$

[4]The LF of a parameter of interest (in this case, the target position) is the probability density function (pdf) of the observation conditioned on the parameter [1]. In the literature, one can find the term "likelihood of the observation," which is incorrect.

[5]The ML estimator is implemented using the LS technique [1].

Table I
Steady-State Bias Error Variances and Standard Deviations for Each Scenario

Scenario	$(1-\alpha_1, 1-\alpha_2)$	(τ_1, τ_2)	P_{11}	$\sigma_{\hat{b}_1}$	P_{22}	$\sigma_{\hat{b}_2}$
1	$(10^{-4}, 10^{-2})$	$(10^3, 10^1)$	0.1673	0.4091	0.3084	0.5553
2	$(10^{-5}, 10^{-2})$	$(10^4, 10^1)$	0.0598	0.2445	0.2220	0.4712
3	$(10^{-4}, 10^{-3})$	$(10^3, 10^2)$	0.3689	0.6074	0.4014	0.6336

In (50), P_{mn} is the (m, n) element of the calculated covariance matrix associated with the bias estimate vector (17).

The variance corresponding to the fused bias-compensated observation (47) is

$$P^{\text{Fbc}} = \left[(H^{\text{Fbc}})'(R^{\text{Fbc}})^{-1}H^{\text{Fbc}}\right]^{-1}. \tag{51}$$

The naïve fusion (with no bias compensation—"Fnbc") is

$$\hat{\xi}^{\text{Fnbc}} = \frac{\sigma_{w_1}^{-2}z_1 + \sigma_{w_2}^{-2}z_2}{\sigma_{w_1}^{-2} + \sigma_{w_2}^{-2}}, \tag{52}$$

which has an MS error

$$P^{\text{Fnbc}} = \frac{\sigma_{w_1}^{-4}(\sigma_{w_1}^2 + \sigma_{\hat{b}_1}^2) + \sigma_{w_2}^{-4}(\sigma_{w_2}^2 + \sigma_{\hat{b}_2}^2)}{(\sigma_{w_1}^{-2} + \sigma_{w_2}^{-2})^2}. \tag{53}$$

Note that the (non-Bayesian) ML fusion technique is the same as the (Bayesian) MMSE fusion technique for dependent tracks (with Gaussian errors), as discussed in [2].

V. SIMULATION RESULTS

Numerical examples are shown in this section with simulation results. For simplicity and illustration, consider the stochastic biases to be estimated have MS values $\sigma_{b_1}^2 = \sigma_{b_2}^2 = 1$ for both sensors. The observation noises $\sigma_{w_1}^2$ and $\sigma_{w_2}^2$ share the same variance 1. The sampling interval $T = 0.1$ s. Simulation results are obtained based on 100 MC runs.

Table II
Calculated Bias Estimate Variances for Various Numbers of Scans for Each Scenario

Scans	Scenario	$(1-\alpha_1, 1-\alpha_2)$	(τ_1, τ_2)	P_{11}	P_{22}	P^{Fbc}
500	1	$(10^{-4}, 10^{-2})$	$(10^3, 10^1)$	0.2529	0.3786	0.7698
	2	$(10^{-5}, 10^{-2})$	$(10^4, 10^1)$	0.2308	0.3620	0.7505
	3	$(10^{-4}, 10^{-3})$	$(10^3, 10^2)$	0.4686	0.4959	0.9662
1000	1	$(10^{-4}, 10^{-2})$	$(10^3, 10^1)$	0.1952	0.3313	0.7171
	2	$(10^{-5}, 10^{-2})$	$(10^4, 10^1)$	0.1512	0.2969	0.6779
	3	$(10^{-4}, 10^{-3})$	$(10^3, 10^2)$	0.4405	0.4692	0.9388
2000	1	$(10^{-4}, 10^{-2})$	$(10^3, 10^1)$	0.1709	0.3113	0.6949
	2	$(10^{-5}, 10^{-2})$	$(10^4, 10^1)$	0.0963	0.2519	0.6277
	3	$(10^{-4}, 10^{-3})$	$(10^3, 10^2)$	0.4062	0.4363	0.9054

Table III
Bias NEES for Each Scenario With Various Numbers of Sampling Scans From 100 Runs; 95% Probability Interval is [1.63 2.41]

Scenario	$N = 500$	$N = 1000$	$N = 2000$
1	1.7321	2.1496	2.1339
2	2.1413	2.0102	1.8412
3	2.0802	2.1724	2.0791

A. Numerical Results

Three scenarios are considered in this section with different values of the pair $(1-\alpha_1, 1-\alpha_2)$: $(10^{-4}, 10^{-2})$, $(10^{-5}, 10^{-2})$, and $(10^{-4}, 10^{-3})$. The corresponding time-constant pairs (τ_1, τ_2) are $(10^3, 10^1)$ s, $(10^4, 10^1)$ s, and $(10^3, 10^2)$ s. With the same sampling interval, a smaller $1-\alpha_i$ (α_i closer to 1) indicates a larger time constant and the corresponding bias has a smoother trajectory [7]. As discussed at the end of Section II, the covariance of the bias estimates will converge to an SS value. Table I lists the solutions of the SS variances from the discrete-time Ricatti equation for each scenario. Note that for such a system with α_i close to 1, the convergence rate is slow due to the small Kalman gain. The calculated covariances (variances for each bias, specifically) for each scenario with respect to various numbers of scans N are shown in Table II. The total simulation duration is NT. The filter consistency is tested using the normalized estimation error squared (NEES) [1], which is Chi-square distributed with the number of degrees of freedom given by the number of MC runs (n_{MC}) and the dimension of the parameter vector (2 in this case). The 95% probability interval with $n_{\text{MC}} = 100$ for the bias estimate is [1.63 2.41]. It can be seen from Table III that the NEES[6] for all the cases falls into the above interval and the estimator is thus consistent, i.e., the actual MSE matches the filter-calculated variance.

In all the cases considered, the estimator reduced the uncertainty of the biases. In Scenario 2 with 1000 scans, the MS value of each bias is 1 before the biases are estimated, which is reduced to 0.1512 for b_1 (61% standard deviation reduction) and 0.2969 for b_2 (46% standard deviation reduction). Similarly, for Scenario 1 with 1000 scans, the standard deviation reduction is 56% for b_1 and 42% for b_2. It can be seen that with 2000 scans (total observation time 200 s), the calculated variance almost reaches its SS value. The performance of the estimator is sensitive to the bias models, i.e., when α_1 and α_2 are close, such as in Scenario 3, the estimation performance (uncertainty reduction) is not as significant due to the marginal observability (the observability matrix $\mathcal{O}$ being nearly singular).

In Table II, the variances of the fused observations (51) for each scenario are also shown. The uncompen-

[6]The number of degrees of freedom is $100 \times 2 = 200$ and the NEES uses division by 100, i.e., it should be around 2.

Table IV
Sensitivity of MSE of Each Bias Estimate ($\mathrm{MSE}_{b_1}, \mathrm{MSE}_{b_2}$) for Various Filter Model Time Constants ($\tau_1^{\mathrm{MOD}}, \tau_2^{\mathrm{MOD}}$) From 1000 Runs With True Model $\tau_1^{\mathrm{TRUE}} = 10^3$ s, $\tau_2^{\mathrm{TRUE}} = 10$ s. Note that Unless $\tau^{\mathrm{MOD}} = \tau^{\mathrm{TRUE}}$ the Filters Are Mismatched

$(\mathrm{MSE}_{b_1}, \mathrm{MSE}_{b_2})$		τ_2^{MOD}			
		$0.5\tau_2^{\mathrm{TRUE}}$	τ_2^{TRUE}	$2\tau_2^{\mathrm{TRUE}}$	$5\tau_2^{\mathrm{TRUE}}$
τ_1^{MOD}	$0.5\tau_1^{\mathrm{TRUE}}$	(0.1971, 0.3391)	(0.1826, 0.3163)	(0.1897, 0.3333)	(0.2358, 0.4155)
	τ_1^{TRUE}	(0.1812, 0.3236)	(0.1760, 0.3109)	(0.1846, 0.3309)	(0.2239, 0.4087)
	$2\tau_1^{\mathrm{TRUE}}$	(0.1792, 0.3194)	(0.1801, 0.3145)	(0.1882, 0.3354)	(0.2228, 0.4102)
	$5\tau_1^{\mathrm{TRUE}}$	(0.1884, 0.3240)	(0.1897, 0.3225)	(0.1947, 0.3423)	(0.2252, 0.4141)

Table V
Sensitivity of the Fused Observation MSE ($\mathrm{MSE}_{\mathrm{Fbc}}$) With Bias Compensation From 1000 Runs

$\mathrm{MSE}_{\mathrm{Fbc}}$		τ_2^{MOD}			
		$0.5\tau_2^{\mathrm{TRUE}}$	τ_2^{TRUE}	$2\tau_2^{\mathrm{TRUE}}$	$5\tau_2^{\mathrm{TRUE}}$
τ_1^{MOD}	$0.5\tau_1^{\mathrm{TRUE}}$	0.7041	0.6878	0.6991	0.7557
	τ_1^{TRUE}	0.6939	0.6863	0.6985	0.7479
	$2\tau_1^{\mathrm{TRUE}}$	0.6956	0.6938	0.7050	0.7492
	$5\tau_1^{\mathrm{TRUE}}$	0.7070	0.7056	0.7136	0.7533

sated sensor observation bias has MS values $\sigma_{b_i}^2 = \sigma_{w_i}^2 = 1$ and the naïvely fused error has an MS value $P^{\mathrm{Fnbc}} = 1$ in all cases. After bias estimation and the compensated fusion, the MS values have been significantly reduced. The MS reduction is up to 37% (with $P^{\mathrm{Fbc}} = 0.63$ for Scenario 2).

B. Sensitivity

In the real world, the true bias models are not available in most cases, which will result in a mismatched filter in estimation. The sensitivity of the proposed estimation method is shown in Table IV, where the MSEs of the bias estimate (obtained through a mismatched filter for Scenario 1) are listed. The true bias dynamic models have time constants $\tau_1^{\mathrm{TRUE}} = 10^3$ s and $\tau_2^{\mathrm{TRUE}} = 10^1$ s. To test the sensitivity of the proposed estimator, different time constants are considered in the KF. The biases' time constants used in the KF are τ_1^{MOD} and τ_2^{MOD}, respectively. The calculated variances for the bias estimates are independent of the actual measurements with $P_{11} = 0.1709$ and $P_{22} = 0.3113$ for all the cases considered. Since the bias dynamic models used in the filter are different from the true ones, the consistency is lost. It can be seen that, if the true bias model has time constants different from those assumed in the estimator, the MSE of the estimate increases. Nevertheless, there is always a reduction in the bias error uncertainty.

The corresponding MSEs of the fused observations[7] with bias compensation for various filter bias models are shown in Table V. The simulation results are obtained from 1000 MC runs. It can be seen that, even with mismatched models in the filter, the fusion of the observations with bias compensation always achieves a smaller MSE than the MSE of the "naïve" fusion, which is 1.

[7]These are MS errors, not covariances, in the case of mismatched filters.

Table VI
Sample Autocorrelation Based Estimation of Bias b_1 Model for Various Data Batch Sizes

Batch length	5×10^5	10^6	10^7
e_1	0.122	0.371	0.621
m_1	1.001	1.000	1.000

C. Bias Model Identification

In this subsection, the scenario with decorrelation true time constants $\tau_1^{\mathrm{TRUE}} = 10^3$ s and $\tau_2^{\mathrm{TRUE}} = 10^1$ s is considered. The two approaches introduced in Section III are tested in the following.

Define the ratio of the estimated time constant to the true constant as

$$e_i = \hat{\tau}_i / \tau_i^{\mathrm{TRUE}}, \qquad i = 1, 2, \tag{54}$$

which indicates the accuracy of the estimation and can be used for sensitivity analysis, and

$$m_i = \hat{\sigma}_{w_i}^2 / (\hat{\sigma}_{w_i}^{\mathrm{TRUE}})^2, \qquad i = 1, 2, \tag{55}$$

to be the ratio of the estimated measurement noise variance and the truth, taken as $(\hat{\sigma}_{w_i}^{\mathrm{TRUE}})^2 = 1$. The simulation results of e_i and m_i from the sample autocorrelation based estimation of the corresponding bias model are shown in Tables VI and VII with various data batch lengths for $i = 1$ and $i = 2$, respectively. It can be seen that, with more data, the accuracy of the model estimation has been better. Note that bias model 1, with a higher time constant, requires a longer batch.

Table VII
Sample Autocorrelation Based Estimation of Bias b_2 Model for Various Data Batch Sizes

Batch length	5×10^4	10^5	5×10^5
e_2	0.650	0.736	0.916
m_2	0.995	0.986	0.999

Table VIII
MLE From 100 MC Runs (Batch Size $L = 250$)

i	$\Delta\alpha$	$\Delta\sigma_w^2$	$1-\alpha$ grid	σ_w^2 grid	$\bar{e}_i$	RMSE(e_i)	$\bar{m}_i$	RMSE(m_i)
1	0.001	0.01	[0.95 0.999]	[0.5 1.5]	5.489	4.611	1.000	0.087
2	0.00001	0.01	[0.999 0.99999]	[0.5 1.5]	1.369	1.265	1.001	0.095

The MLE is obtained via numerical search, where the search intervals for time constants and measurement noises as well as the grid steps are shown in Table VIII. The observation batch size is $L = 250$ and the simulation results [mean and variance of the ratios, (54) and (55), respectively] are based on 100 MC runs. The MSEs of bias estimation and fusion with bias compensation based on the result of system identification are are shown in Table IX for the two approaches discussed. The degradation of $\text{MSE}_{\text{FbcMI}}$ (MSE of fusion with bias compensation based on model identification) versus MSE_{Fbc} (MSE from correct filter) is around 11% for the autocorrelation-based model identification, and is around 6% for ML-based identification due to the estimated model in the filter. Note that the MSE of fused observation is the most important, and, even with a relatively high error in the bias model estimation, the fused MSE is clearly better with bias compensation than without. Also note that the ML procedure requires a much shorter batch length.

VI. CONCLUSIONS

In this work, the bias estimation for collocated synchronized sensors is solved using a target of opportunity. The sensor biases are slow varying, modeled as OU processes. Only the difference between the sensor observations is used for the bias estimation, which is thus independent of the target-state estimation. The system is observable when the biases have different eigenvalues in their noise-driven discrete time dynamic models. The bias model parameters can be obtained directly via sample autocorrelations or via ML estimation. With bias estimates and by accounting for the residual biases, the fusion of the bias-compensated observations is carried out under the ML criterion. The standard deviation of the fused measurement is significantly reduced. The performances, in terms of both accuracy and convergence time, are sensitive to and depend on the bias dynamics. The bias-estimation consistency is proved via simulation results. Using the ML criterion, the fusion of the observations carried out with bias compensation results in a significant MSE reduction for the fused measurement. The proposed algorithm is also shown to provide benefits even with mismatched filers with bias dynamic models different from the true ones.

Table IX
MSE of Fused Observation Based on System Identification From 100 Runs

Bias estimation approach	MSE_{b_1}	MSE_{b_2}	$\text{MES}_{\text{FbcMI}}$
Autocorrelation based (batch length 5×10^6)	0.2278	0.3416	0.7728
ML (batch length 250)	0.1650	0.3351	0.7378

APPENDIX

A LEAST SQUARES ESTIMATOR OF BIAS MODEL PARAMETERS

Given the sample autocorrelations $\{r(1), r(2), \ldots, r(M))\}$, the LS estimates are

$$\begin{bmatrix}\hat{\beta}\\ \hat{\gamma}\end{bmatrix} = \arg\min_{\beta,\gamma} E(\beta,\gamma) = \arg\min_{\beta,\gamma}\left\{\sum_{m=1}^{M}[\ln r(m) - (m\beta+\gamma)]^2\right\}. \quad \text{(A1)}$$

Minimization of the above expression

$$\frac{\partial E(\beta,\gamma)}{\partial\beta} = 0,\ \frac{\partial E(\beta,\gamma)}{\partial\gamma} = 0, \quad \text{(A2)}$$

where

$$\frac{\partial E(\beta,\gamma)}{\partial\beta} = \sum_{m=1}^{M} -2[\ln r(m) - (m\beta+\gamma)]m, \quad \text{(A3)}$$

$$\frac{\partial E(\beta,\gamma)}{\partial\gamma} = \sum_{m=1}^{M} -2[\ln r(m) - (m\beta+\gamma)]. \quad \text{(A4)}$$

Setting $\frac{\partial E(\beta,\gamma)}{\partial\beta} = \frac{\partial E(\beta,\gamma)}{\partial\gamma} = 0$ gives

$$\sum_{m=1}^{M}[\ln r(m) - (m\beta+\gamma)]m = 0, \quad \text{(A5)}$$

$$\sum_{m=1}^{M}[\ln r(m) - (m\beta+\gamma)] = 0. \quad \text{(A6)}$$

Rewriting the above equations as

$$\left(\sum_{m=1}^{M} m^2\right)\beta + \left(\sum_{m=1}^{M} m\right)\gamma = \sum_{m=1}^{M} m\ln r(m), \quad \text{(A7)}$$

$$\left(\sum_{m=1}^{M} m\right)\beta + \left(\sum_{m=1}^{M} 1\right)\gamma = \sum_{m=1}^{M} \ln r(m), \quad \text{(A8)}$$

the LS estimates are (the solution to these equations)

$$\begin{bmatrix} \hat{\beta} \\ \hat{\gamma} \end{bmatrix} = \begin{bmatrix} \sum_{m=1}^{M} m^2 & \sum_{m=1}^{M} m \\ \sum_{m=1}^{M} m & \sum_{m=1}^{M} 1 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{m=1}^{M} m \ln r(m) \\ \sum_{m=1}^{M} \ln r(m) \end{bmatrix}. \tag{A9}$$

REFERENCES

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

[2] Y. Bar-Shalom, P. K. Willett, and X. Tian
Tracking and Data Fusion. Storrs, CT, USA: YBS Publishing, 2011.

[3] Y. Bar-Shalom
"Airborne GMTI radar position bias estimation using static-rotator targets of opportunity,"
IEEE Trans. Aerosp. Electron. Syst., vol. 37, no. 2, pp. 695–699, Apr. 2001.

[4] P. D. Burns and W. D. Blair
"Sensor bias estimation from measurements of known trajectories,"
in *Proc. 37th Southeastern Symp. Syst. Theory*, Tuskegee, AL, USA, 2005.

[5] P. Braca, L. Millefiori, K. Bryan, and P. Willett
"Stochastic mean-reverting process for long-term prediction,"
IEEE Trans. Aerosp. Electron. Syst., vol. 52, no. 5, pp. 2313–2330, Sep. 2016.

[6] J. C. G. Franco
"Maximum-likelihood estimation of mean reverting processes," Onward, Inc., Working paper, 2003.

[7] K. Kastella and B. Yeary
"Bias modeling and estimation for GMTI applications,"
in *Proc. 3rd Intl. Conf. Inf. Fusion*, Paris, France, Jul. 2000.

[8] M. Kowalski, Y. Bar-Shalom, P. Willett, D. Belfadel, and F. Daum
"Bias CRLB in sine space for a 3-dimensional sensor,"
IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 1, pp. 673–686, Feb. 2020.

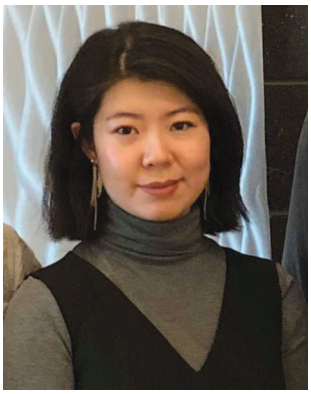
[9] T. Kailath
Linear Systems. Englewood Cliffs, NJ, USA: Prentice-Hall, 1980.

[10] X. Lin, Y. Bar-Shalom, and T. Kirubarajan
"Exact multi-sensor dynamic bias estimation with local tracks,"
IEEE Trans. Aerosp. Electron. Syst., vol. 40, no. 2, pp. 576–590, Apr.2004.

[11] X. Lin and Y. Bar-Shalom
"Multisensor target tracking performance with bias compensations,"
IEEE Trans. Aerosp. Electron. Syst., vol. 42, no. 3, pp. 1139–1149, Jul. 2006.

[12] L. M. Millefiori, P. Braca, K. Bryan, and P. Willett
"Modeling vessel kinematics using a stochastic mean-reverting process for long-term prediction,"
IEEE Trans. Aerosp. Electron. Syst., vol. 52, no. 5, pp. 2313–2330, Oct. 2016.

[13] L. M. Millefiori, P. Braca, and P. Willett
"Consistent estimation of randomly sampled Ornstein–Uhlenbeck process long-run mean for long-term target state prediction",
IEEE Signal Process. Lett., vol. 23, no. 11, pp. 1562–1566, Nov. 2016.



Kaipei Yang received the B.S. degree from Northwestern Polytechnical University in 2014 and the Ph.D. degree from the University of Connecticut in 2019. She is now an Assistant Research Professor with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA. Her research interests include statistical signal processing, estimation theory, and information fusion. She gained experience in autonomous driving vehicles while working at NIO in San Jose, CA, USA, in 2018.

Yaakov Bar-Shalom received the B.S. and M.S. degrees from the Technion in 1963 and 1967, respectively, and the Ph.D. degree from Princeton University in 1970, all in electrical engineering. From 1970 to 1976, he was with the Systems Control, Inc., Palo Alto, CA, USA. Currently, he is a Board of Trustees Distinguished Professor with the Department of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering with the University of Connecticut. His current research interests include estimation theory, target tracking, and data fusion. He has published more than 550 papers and book chapters. He coauthored/edited eight books, including *Tracking and Data Fusion* (YBS Publishing, 2011). He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He served as an Associate Editor for the IEEE Transactions on Automatic Control and *Automatica*. He was General Chairman of the 1985 ACC. He served as the Chairman of the Conference Activities Board of the IEEE CSS and a member of its Board of Governors. He served as the General Chairman of FUSION 2000, President of ISIF in 2000 and 2002, and Vice President for Publications during 2004–2013. In 1987, he received the IEEE CSS Distinguished Member Award. Since 1995, he has been a Distinguished Lecturer of the IEEE AESS. He is a co-recipient 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, he was awarded 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.

Peter Willett received the B.A.Sc. (engineering science) degree from the University of Toronto in 1982 and the Ph.D. degree from Princeton University in 1986. Since 1986, He has been a faculty member with the Electrical and Computer Engineering Department at the University of Connecticut. Since 1998, he has been a Professor, and since 2003, an IEEE Fellow. His primary areas of research have been statistical signal processing, detection, machine learning, communications, data fusion, and tracking. He has published more than 650 papers on these topics. He was the Editor-in-Chief for the IEEE Signal Processing Letters from 2014 to 2016. He was the Editor-in-Chief for the IEEE Transactions on Aerospace and Electronic Systems from 2006 to 2011, and then the AESS Vice President for Publications (2012–2014). He was a member of the IEEE AESS Board of Governors (2005–2010, 2011–2016) and of the IEEE Signal Processing Society's Sensor-Array and Multichannel (SAM) technical committee (and Chair 2015–2016).

Hiroshi Inou received the B.E. and M.E. degrees from Tohoku University, Sendai, Japan, in 2001 and 2003, respectively, and the Ph.D. degree from Kobe University, Kobe, Japan, in 2013. Currently, he is the General Manager of Global R&D Tokyo, Denso Corporation, working on research and development for automated/autonomous driving perception, path planning, vehicle motion control, and integration of complex systems. His research covers the area of advanced control of vehicles and theoretical development of nonlinear optimal control, and its implementation in real vehicles. His current interests are robust perception using AI and model-based technology to satisfy automotive grade requirements.

Heterogeneous and Asynchronous Information Matrix Fusion

KAIPEI YANG
YAAKOV BAR-SHALOM
KUO-CHU CHANG

The Information Matrix Fusion (IMF) algorithm for nonlinear, asynchronous (with arbitrary local tracker sampling times for full rate as well as reduced-rate communication) and heterogeneous systems is presented. The heterogeneous estimates from local trackers are in different state spaces with different dimensions and are related by a nonlinear and noninvertible transformation. The main application of these results is the fusion of tracks from radar and infrared/electrooptical sensors. Different from Track-to-Track Fusion, the IMF does not require the cross-covariance between the local estimation errors. The performance of the proposed algorithm is shown via simulation based on Monte Carlo runs and is compared with the optimal solution—full-rate centralized fusion for both full-rate fusion and reduced-rate fusion for heterogeneous and asynchronous sensors.

Manuscript received July 1, 2020; revised July 7, 2020; released for publication October 10, 2020.

Associate Editor: Florian Meyer.

K. Yang and Y. Bar-Shalom are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06268 USA (E-mail: kaipei.yang@uconn.edu, yaakov.bar-shalom@uconn.edu). K.-C. Chang is with the Department of Systems Engineering and Operations Research, George Mason University, Fairfax, VA 22030 USA (E-mail: kchang@gmu.edu).

I. INTRODUCTION

A sensor configuration with complementary sensors at different locations (sensor network) is required in most of the tracking systems to achieve the necessary dependability and estimation accuracy. The best target-state-estimation performance is obtained by a centralized tracker/fuser (CTF), by directly sending to the fusion center (FC) all the measurements of the local sensors. However, CTF is not always available due to the communication constraints in practical situations. In this case, local sensors are capable of performing target-state tracking with their information processing systems. Such a system has a number of tracks that are sent to the FC. High-level algorithms such as Track-to-Track Fusion (T2TF) and Information Matrix Fusion (IMF) are commonly used for their modularity, practicality and scalability.

The IMF algorithm, as derived in [4] and originally presented in [6] for the case where all the local sensors are synchronized, is restricted to (i) linear systems, and (ii) the local trackers (LTs) estimate the same state. This algorithm belongs to the class of Track-to-Track Fusion with Memory (T2TFwM) and is, if operating at a "full communication rate," algebraically equivalent to the Configuration IV (centralized) tracker for linear systems. Unlike the Track-to-Track Fusion without Memory (T2TFwoM) [4], the IMF algorithm does not need the cross-covariance between the track estimation errors.

Asynchronous fusion was considered in [9] for the case where the local trackers estimate the same state. Aeberhard et al. [2] considered an asynchronous IMF with applications for driver-assistance systems. Heterogeneous fusion was investigated in [10], where it was shown how the linear minimum mean-square error (LMMSE) estimator and an (approximate) equivalent measurement based on a lower dimension local-state estimate can be used to update a higher dimension state estimate when these states are related by a nonlinear transformation. However, the fusion algorithm from [10] did not account for the common process noise. A modification of the result of [10] was given in [1] by using the unscented transform to evaluate the necessary covariances. The recent work of Mallick et al. [7] considered the problem of track fusion from heterogeneous sensors (with sampling intervals of the radar a multiple of the interval of the infrared/electrooptical (IR/EO) sensor) by augmenting the lower dimensional state of the IR/EO sensor with a number of range estimates based on a priori information, thus making it of the same dimension as the radar's state estimate. The work [11] derived the relationship between the process noise covariance matrices of the two state vectors (to account for the common process noise) and provided the expression of the covariance matrix of the heterogeneous estimation errors, which is needed in T2TF.

In [12], the IMF algorithm was first derived for nonlinear systems where the local filters are extended

Kalman filters (EKFs), rather than Kalman filters (KFs), estimating the same state and the conditions under which it holds. The fusion equations for the asynchronous case—arbitrary LT sensor sampling times—with LT-driven communication as well as FC-driven communication are also given for a system with homogeneous sensors.

In the present work, which is an extension of [12], the IMF algorithm is generalized to asynchronous heterogeneous systems where the local filters estimate different states (in different spaces of different dimensions), related by a nonlinear transformation, and the local sensors are running at different sampling rates. Such a situation occurs when the first tracker, using radar data, estimates the full Cartesian state of the target, while the second tracker, using a passive sensor (e.g., IR/EO), estimates the (lower dimension) angular state of the target. These two different-dimension states are related by a nonlinear transformation with no inverse. It is shown that one can combine state estimates of different dimensions in the IMF algorithm by constructing a "mapped" information involving the Jacobian of the nonlinear transformation that relates these states. Specifically, the IMF algorithm for an asynchronous case is derived for both an LT (full-rate) driven case and an FC (reduced-rate) driven case. This is investigated since the asynchronous case is motivated by the real-world scenario where the sensors have different sampling frequencies. In this work, we considered one-way communication from LT to FC without feedback. Note that there is no communication delay. In other words, the time index used later in the IMF formulas represents the LT sensor observation time, LT update time, as well as FC fusion time for the synchronous case. For the asynchronous case, zero communication delay implies that there is no out-of-order information received at FC from LT.

Section II presents the IMF algorithm for nonlinear filters with homogeneous sensors, i.e., the state estimates are in the same state space. Section III introduces the heterogeneous system in detail. The IMF algorithm for synchronous heterogeneous sensors is shown in Section IV . The asynchronous IMF for heterogeneous sensors is discussed in Section V for both the LT-driven case and the FC-driven case. Section VI presents the simulation results of an asynchronous heterogeneous IMF and compares them with the (i) optimal solution—centralized tracking/fusion and (ii) T2TF solution. Conclusions are presented in Section VII. Notations used in equations are summarized in Table I.

Table I
Notations Used for IMF Algorithm

Indices:	
t_k	Times when the FC carries out fusion
k	Discrete time index
T	Sampling interval
i	Sensor index
$t^r(t_k)$	Most recent times prior to t_k at which LT r sent information to the FC
$t^e(t_k)$	Most recent times prior to t_k at which LT e sent information to the FC
t^e_m	Sampling time of EO sensor with index m
t^r_l	Sampling time of radar with index l
Parameters:	
n^e	Dimension of state estimate from EO tracker
n^r	Dimension of state estimate from radar tracker
N_s	Number of local sensors
Variables:	
z^i	Measurement from sensor i
$\hat{x}^i$	State estimate of track i, used in homogeneous IMF
P^i	Covariance corresponding to $\hat{x}^i$, used in homogeneous IMF
$\hat{x}^e$, P^e	Estimate from EO sensor with dimension n^e and corresponding covariance
$\hat{x}^r$, P^r	Estimate from radar with dimension n^r and corresponding covariance
$\hat{x}^E$, P^E	Estimate obtained using the $\hat{x}^e$ with dimension n^r and corresponding covariance
$\hat{y}^E(k\|k)$	Mapped information state
$\hat{y}^E(k\|k-1)$	Mapped predicated information state

II. NONLINEAR INFORMATION MATRIX FUSION

The IMF with nonlinear filters for homogeneous sensors, i.e., the states are in the same space, is discussed in detail in [12] for both the synchronous case and the asynchronous case. The derivations will not be repeated here for the sake of brevity. In this section, it is assumed that each local filter/LT uses the same target state model with an EKF as the tracker, and they are synchronized.

Under full-rate communication, each LT communicates to the FC its updates as they are obtained and the FC then updates its fused state.

The LT-state update at sensor i at time t_k (indicated in the sequel by its index only) is given by [4]

$$\hat{x}^i(k|k) = \hat{x}^i(k|k-1) + P^i(k|k)H^i[k, \hat{x}^i(k|k-1)]^{\mathrm{T}} \cdot R^i(k)^{-1}\left[z^i(k) - h^i[k, \hat{x}^i(k|k-1)]\right] \quad (1)$$

using the measurements

$$z^i(k) = h^i[k, x(k)] + w^i(k), \quad (2)$$

where $w^i(k)$ is the zero-mean white measurement noise with covariance $R^i(k)$ and h^i is its measurement function, with Jacobian

$$H^i[k, \hat{x}^i(k|k-1)] \triangleq \left[\nabla_{x(k)} h^i[k, x(k)]^{\mathrm{T}}\right]^{\mathrm{T}}\Big|_{x(k)=\hat{x}^i(k|k-1)}. \quad (3)$$

The covariance-update equation in the information matrix form is (see [3, eq. (5.2.3-16)])

$$P^i(k|k)^{-1} = P^i(k|k-1)^{-1} + H^i[k, \hat{x}^i(k|k-1)]^{\mathrm{T}} \cdot R^i(k)^{-1} H^i[k, \hat{x}^i(k|k-1)]. \quad (4)$$

The EKF information-state update at sensor i is [11]

$$\begin{aligned} &P^i(k|k)^{-1}\hat{x}^i(k|k) \\ &= P^i(k|k-1)^{-1}\hat{x}^i(k|k-1) + H^i[k,\hat{x}^i(k|k-1)]^{\mathrm{T}} \\ &\quad \cdot R^i(k)^{-1}\left[z^i(k) - h^i[k,\hat{x}^i(k|k-1)]\right] \\ &\quad + H^i[k,\hat{x}^i(k|k-1)]\hat{x}^i(k|k-1), \end{aligned} \tag{5}$$

which gives (by rearranging)[1]

$$\begin{aligned} &H^i[k,\hat{x}^i(k|k-1)]^{\mathrm{T}}R^i(k)^{-1}\left[z^i(k) - h^i[k,\hat{x}^i(k|k-1)]\right] \\ &\quad + H^i[k,\hat{x}^i(k|k-1)]\hat{x}^i(k|k-1) \\ &\quad = P^i(k|k)^{-1}\hat{x}^i(k|k) - P^i(k|k-1)^{-1}\hat{x}^i(k|k-1). \end{aligned} \tag{6}$$

The *full-rate information-state fusion* equation is

$$\begin{aligned} P(k|k)^{-1}\hat{x}(k|k) = P(k|k-1)^{-1}\hat{x}(k|k-1) \\ + \sum_{i=1}^{N_s}\left[P^i(k|k)^{-1}\hat{x}^i(k|k)\right. \\ \left. - P^i(k|k-1)^{-1}\hat{x}^i(k|k-1)\right], \end{aligned} \tag{7}$$

where N_s is the number of sensors. The differences of the predicted and updated information states in the summation above are the "new information" from each of the sensors. This new information is exactly equivalent to the innovation in the KF in the linear case—see (6)—and, thus, it is uncorrelated with the past information.

The *information matrix fusion* equation is

$$P(k|k)^{-1} = P(k|k-1)^{-1} + \sum_{i=1}^{N_s}[P^i(k|k)^{-1} - P^i(k|k-1)^{-1}], \tag{8}$$

i.e., the same as in the linear case but subject to the approximations (linearization).

III. HETEROGENEOUS STATES

The IMF algorithm, as shown in (7), requires the addition of the information vectors $(P^i)^{-1}\hat{x}^i$ across the local trackers, i.e., they have to have the same dimension. In the case where one sensor is a radar (with the corresponding target estimate of dimension n^r and its covariance, superscripted by r) and the other sensor is an IR/EO one (with the corresponding target estimate of dimension n^e and its covariance, superscripted by e), the two estimated state vectors have different dimensions (they are in different spaces). Consequently, the corresponding new information, based on the different-dimension local information states, cannot be added as required by (7).

The smaller dimension (n^e) state is related to the higher dimension (n^r) state according to

$$x^e = g(x^r, p^r, p^e), \tag{9}$$

where the time arguments are omitted for simplicity, and p^r and p^e are the position vectors of the radar and the EO sensor, respectively. Since $g(\cdot)$ maps the n^r-dimensional vector x^r to the (lower) n^e-dimensional vector x^e, it is not invertible, i.e., one cannot obtain an estimate of the full Cartesian state (of dimension n^r) based on the angular state estimate from the EO sensor.

Consider the estimate $\hat{x}^e$ from the EO sensor (local track) as an "observation" z of the n^r-dimensional state vector x^r (truth) related by

$$z = g(x^r) + w, \tag{10}$$

where w is a zero-mean noise with covariance matrix P^e. The solution (desired estimate) of the above system is defined as $\hat{x}^E$ (of dimension n^r), where the superscript E indicates the estimate obtained using the observation $\hat{x}^e$ through the nonlinear relationship $g(\cdot)$.[2] The (hypothetical[3]) least-squares (LS) estimate of dimension n^r based on (10) (following [3, eq. (3.4.4-11)], using the radar estimate $\hat{x}^r$ since the true state is not available, is given by

$$\hat{x}^E = (G'R^{-1}G)^{-1}G'R^{-1}(z - h(\hat{x}^r)) + \hat{x}^r \tag{11}$$

and the covariance corresponding to $\hat{x}^E$ is

$$P^E = (G'R^{-1}G)^{-1}, \tag{12}$$

where G is the Jacobian evaluated at $\hat{x}^r$ and

$$G(\hat{x}^r) \triangleq \left[\nabla_{x^r} g[x^r]^{\mathrm{T}}\right]^{\mathrm{T}}\Big|_{x^r=\hat{x}^r} \tag{13}$$

is the $(n^e \times n^r)$ Jacobian. Note that $G'R^{-1}G$ is not invertible because the right-hand side of (12) has rank $n^e < n^r$, and the covariance matrix does not exist. In this case, one cannot obtain an estimate of the full Cartesian state (of dimension n^r) based on the angular-state estimate from the EO sensor. However, the above equations provide the motivation for the following implementable algorithm that can overcome the unequal-state-dimension problem: Instead of the "mapped" estimate (11) and covariance matrix (12), one can calculate the "mapped new information" directly, which is what the IMF equations need. The following approach is taken to overcome the incompatibility of state dimensions and singularity of (12).

IV. HETEROGENEOUS IMF FOR SYNCHRONOUS CASE

Define an n^r-dimensional "mapped (from the EO/IR state space to radar state space) information state"

$$\hat{y}^E(k|k) \triangleq P^E(k|k)^{-1}\hat{x}^E(k|k) \tag{14}$$

and a "mapped predicted information state"

$$\hat{y}^E(k|k-1) \triangleq P^E(k|k-1)^{-1}\hat{x}^E(k|k-1) \tag{15}$$

[1] Note that equations (5) and (6) in [12] have typos and the correct ones are, respectively, given in (5) and (6) in this paper.

[2] This is (9) omitting the sensor positions for simplicity.

[3] To be defined in the sequel—this estimate cannot be obtained since $n^e < n^r$.

in order to obtain the "mapped new information" $\hat{y}^E(k|k) - \hat{y}^E(k|k-1)$. Although $\hat{y}^E(k|k)$ and $\hat{y}^E(k|k-1)$ cannot be obtained since $\hat{x}^E$ and P^E are not available, they are not necessary and *only their difference*—the "mapped new information"—will be needed in the information-state-fusion equation, as it will be shown in the sequel.

The state update equation for (10) with the time arguments, based on (1), can be written as

$$\hat{x}^E(k|k) = \hat{x}^E(k|k-1) + P^E(k|k)G[k,\hat{x}^E(k|k-1)]^{\mathrm{T}} \cdot P^e(k|k)^{-1}\left[\hat{x}^e(k|k) - g[k,\hat{x}^E(k|k-1)]\right], \tag{16}$$

where

$$G[k,\hat{x}^E(k|k-1)] \triangleq \left[\nabla_x g[x]^{\mathrm{T}}\right]^{\mathrm{T}}\Big|_{x=\hat{x}^E(k|k-1)} \tag{17}$$

is the $(n^e \times n^r)$ Jacobian, the evaluation of which is discussed in the sequel. The covariance update equation in the information matrix form, based on (4), is

$$P^E(k|k)^{-1} = P^E(k|k-1)^{-1} + G[k,\hat{x}^E(k|k-1)]^{\mathrm{T}} \cdot P^e(k|k)^{-1}G[k,\hat{x}^E(k|k-1)]. \tag{18}$$

The "mapped new information" from the sensor of dimension n^e into the space of dimension $n^r > n^e$ can be obtained, by substituting (16) and (18) into (14) and (15) and following (6), as[4]

$$\begin{aligned}\hat{y}^E(k|k) - \hat{y}^E(k|k-1) &= G[k,\hat{x}^E(k|k-1)]^{\mathrm{T}}P^e(k|k)^{-1} \cdot \left\{\hat{x}^e(k|k) - g[k,\hat{x}^E(k|k-1)] + G[k,\hat{x}^E(k|k-1)]\hat{x}^E(k|k-1)\right\} \qquad (19)\\ &\approx G[k,\hat{x}(k|k-1)]^{\mathrm{T}}P^e(k|k)^{-1} \left\{\hat{x}^e(k|k) - g[k,\hat{x}(k|k-1)] + G[k,\hat{x}(k|k-1)]\hat{x}(k|k-1)\right\}, \qquad (20)\end{aligned}$$

where the approximate equality above is obtained by the substitution $\hat{x}^E \to \hat{x}$. The evaluation of the Jacobian (17) and state prediction in (19) can be done using the FC estimate $\hat{x}$ rather than $\hat{x}^E$ (since the latter is not available). In this case, we use $G[k,\hat{x}(k|k-1)]$ and $g[k,\hat{x}(k|k-1)]$ instead of $G[k,\hat{x}^E(k|k-1)]$ and $g[k,\hat{x}^E(k|k-1)]$, respectively. Therefore, (20) will be used to obtain the *synchronous heterogeneous information state fusion equation* with full-rate communication by modifying (7) as follows:

$$\begin{aligned}P(k|k)^{-1}\hat{x}(k|k) &= P(k|k-1)^{-1}\hat{x}(k|k-1) + \left\{P^r(k|k)^{-1}\hat{x}^r(k|k) - P^r(k|k-1)^{-1}\hat{x}^r(k|k-1)\right\} + \left\{\hat{y}^E(k|k) - \hat{y}^E(k|k-1)\right\}. \qquad (21)\end{aligned}$$

Note that the entire right-hand side of (21) has dimension n^r, i.e., the problem of unequal state dimensions has been eliminated. The corresponding *synchronous heterogeneous information matrix fusion equation*, based on (18) and (8), is

$$\begin{aligned}P(k|k)^{-1} &= P(k|k-1)^{-1} + \left\{P^r(k|k)^{-1} - P^r(k|k-1)^{-1}\right\} + \left\{P^E(k|k)^{-1} - P^E(k|k-1)^{-1}\right\}\\ &= P(k|k-1)^{-1} + \left\{P^r(k|k)^{-1} - P^r(k|k-1)^{-1}\right\} + \left\{G[k,\hat{x}^E(k|k-1)]^{\mathrm{T}}P^e(k|k)^{-1}G[k,\hat{x}^E(k|k-1)]\right\}. \qquad (22)\end{aligned}$$

At initialization, one needs the radar's (full-state) estimate to evaluate the Jacobian G.

The fusion architecture for a synchronous heterogeneous IMF is shown in Fig. 1, where the dashed circle indicates the mapping of the new information from angle space to Cartesian space.

V. HETEROGENEOUS IMF FOR ASYNCHRONOUS CASE

A. LT-Driven Asynchronous Case

With LT/local filter-driven communication, the fusion in an asynchronous system (e.g., with tracks from radar and IR/EO sensors) is carried out whenever the FC receives new information. In this case, the system is updated with full rate. As shown in Fig. 2, sensor r is assumed to be the active one (radar) with the state vector in the larger state space (of dimension $n^r_{\mathbf{x}}$) and sensor e is the passive EO/IR with the state vector in the smaller state space (of dimension $n^e_{\mathbf{x}} < n^r_{\mathbf{x}}$). For the FC, the fusion times are equal to the times when new information is obtained. From Fig. 2, we have

$$t_k = t^e_m \tag{23}$$

and, with $l \ni t^r_l < t_k$,

$$t_{k-1} = \max\{t^r_l, t^e_{m-1}\}, \tag{24}$$

where l and m denote the respective LT sampling indices. LT-driven asynchronous fusion (full rate) is carried out whenever an LT has new information delivered to the FC.

The "mapped new information," based on (20), is

$$\begin{aligned}\hat{y}^E(t_k|t_k) - \hat{y}^E[t_k|t^e(t_k)] &= G\big[t_k,\hat{x}^E[t_k|t^e(t_k)]\big]^{\mathrm{T}}P^e(t_k|t_k)^{-1} \cdot \left\{\hat{x}^e(t_k|t_k) - g\big[t_k,\hat{x}^E[t_k|t^e(t_k)]\big] + G\big[t_k,\hat{x}^E[t_k|t^e(t_k)]\big]\hat{x}^E[t_k|t^e(t_k)]\right\}. \qquad (25)\end{aligned}$$

[4] In (6), we substitute $H^i \to G, R^i \to P^e, z^i \to \hat{x}^e, h^i \to g$, and $\hat{x}^i \to \hat{x}^E$.

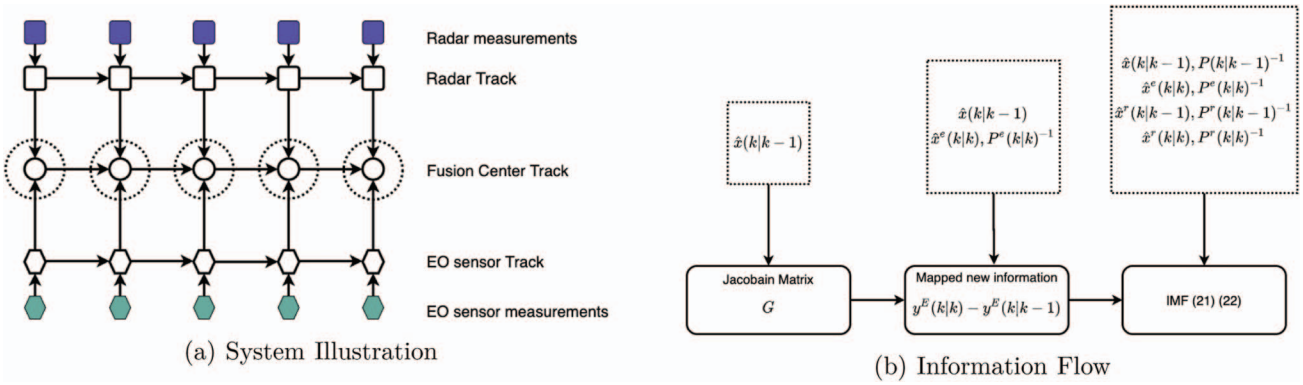


Fig. 1. Fusion architecture for synchronous heterogeneous IMF.

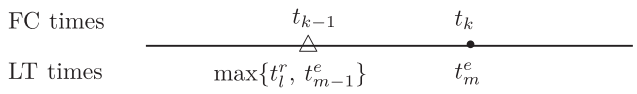


Fig. 2. FC times and LT times in asynchronous IMF—LT driven, update with latest information from sensor e.

Based on the above and discussion about the asynchronous but homogeneous system in [12], the asynchronous heterogeneous information state fusion equation becomes

$$\begin{aligned} &P(t_k|t_k)^{-1}\hat{x}(t_k|t_k) \\ &\quad = P(t_k|t_{k-1})^{-1}\hat{x}(t_k|t_{k-1}) \\ &\qquad +\left\{P^r(t_k|t_k)^{-1}\hat{x}^r(t_k|t_k)\right. \\ &\qquad -P^r[t_k|t^r(t_k)]^{-1}\hat{x}^r[t_k|t^r(t_k)]\Big\}\chi^r(t_k) \\ &\qquad +\left\{\hat{y}^E(t_k|t_k) - \hat{y}^E[t_k|t^e(t_k)]\right\}\chi^e(t_k), \end{aligned} \tag{26}$$

where $t^r(t_k)$ and $t^e(t_k)$ are the most recent times prior to t_k at which LT r and LT e have sent information to the FC (its previous communication), respectively, and $\chi^r(k)$ and $\chi^e(k)$ are the communication indicator functions for LTs,

$$\chi^r(k) = \begin{cases} 1 & \text{if LT } r \text{ sends information to FC at } t_k \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

and

$$\chi^e(k) = \begin{cases} 1 & \text{if LT } e \text{ sends information to FC at } t_k \\ 0 & \text{otherwise} \end{cases}. \tag{28}$$

Note that if $\chi^e(t_k) = 1$ and $\chi^r(t_k) = 0$, (26) carries out the update with the latest information only from sensor e, as illustrated in Fig. 2. If $\chi^e(t_k) = 0$ and $\chi^r(t_k) = 1$, then (26) carries out the update with the latest information only from sensor r.

The corresponding information matrix fusion equation is (modifying (22)) given by

$$\begin{aligned} P(t_k|t_k)^{-1} &= P(t_k|t_{k-1})^{-1} \\ &\quad +\left\{P^r(t_k|t_k)^{-1} - P^r[t_k|t^r(t_k)]^{-1}\right\}\chi^r(t_k) \\ &\quad +\left\{G\big[t_k, \hat{x}^E[t_k|t^e(t_k)]\big]^{\mathrm{T}}P^e(t_k|t_k)^{-1}\right. \\ &\quad \cdot G\big[t_k, \hat{x}^E[t_k|t^e(t_k)]\big]\Big\}\chi^e(t_k). \end{aligned} \tag{29}$$

To implement the above method, the approximations

$$G\big[t_k, \hat{x}^E[t_k|t^e(t_k)]\big] \approx G\big[t_k, \hat{x}[t_k|t^e(t_k)]\big] \tag{30}$$

and

$$\hat{x}^E[t_k|t^e(t_k)] \approx \hat{x}[t_k|t^e(t_k)] \tag{31}$$

are used in (25), (26), and (29).

The fusion architecture for an LT-driven asynchronous heterogeneous IMF is shown in Fig. 3, where the dashed circle indicates the mapping of the new information from angle space to Cartesian (25).

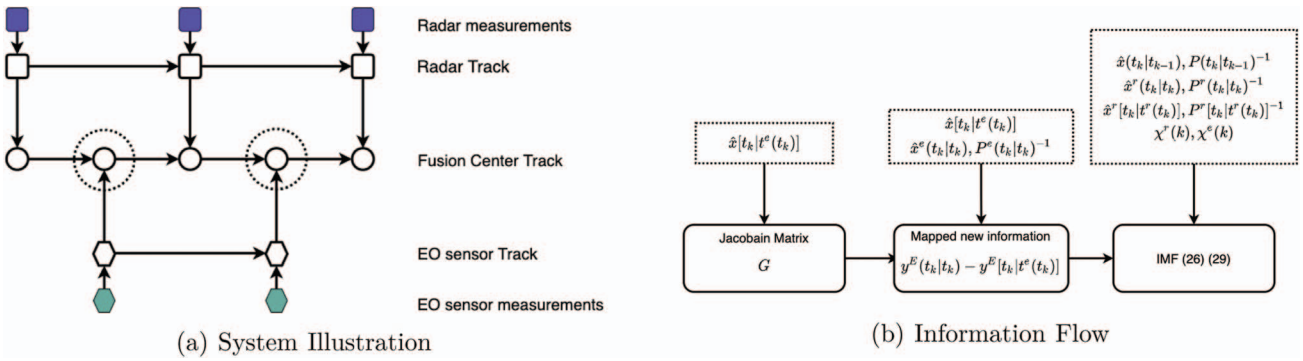


Fig. 3. Fusion architecture for LT-driven asynchronous heterogeneous IMF.

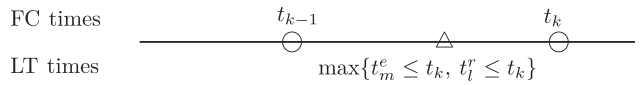


Fig. 4. FC times and LT times in asynchronous IMF—FC driven.

B. FC-Driven Asynchronous Case

In this case, it is assumed that the FC updates its state at intervals τ_k (length of the fusion window); namely, if the current update is at t_k, the previous update was at $t_k - \tau_k = t_{k-1}$. The fusion window ending at t_k is thus the semiclosed interval $(t_k - \tau_k, t_k]$ of length τ_k, which is at the discretion of the FC. The system is updated with a reduced rate. The fusion times at the FC and the latest LT sampling times are shown in Fig. 4.

The information state fusion with FC-driven communication is

$$\begin{aligned} &P(t_k|t_k)^{-1}\hat{x}(t_k|t_k) \\ &\quad = P(t_k|t_{k-1})^{-1}\hat{x}(t_k|t_{k-1}) \\ &\qquad +\{P^r[t_k|t^r(t_k)]^{-1}\hat{x}^r[t_k|t^r(t_k)] \\ &\qquad -P^r[t_k|t^r(t_{k-1})]^{-1}\hat{x}^r[t_k|t^r(t_{k-1})]\} \\ &\qquad +\{\hat{y}^E[t_k|t^e(t_k)] - \hat{y}^E[t_k|t^e(t_{k-1})]\} \end{aligned} \tag{32}$$

and the counterpart of (25) is

$$\begin{aligned} &\hat{y}^E[t_k|t^e(t_k)] - \hat{y}^E[t_k|t^e(t_{k-1})] \\ &\quad = G\big[t_k, \hat{x}^E[t_k|t^e(t_{k-1})]\big]^{\mathrm{T}} P^e[t_k|t^e(t_k)]^{-1} \\ &\qquad \cdot\big\{\hat{x}^e[t_k|t^e(t_k)] - g\big[t_k, \hat{x}^E[t_k|t^e(t_{k-1})]\big] \\ &\qquad + G\big[t_k, \hat{x}^E[t_k|t^e(t_{k-1})]\big]\hat{x}^E[t_k|t^e(t_{k-1})]\big\} \end{aligned} \tag{33}$$

$$\begin{aligned} &\approx G\big[t_k, \hat{x}[t_k|t^e(t_{k-1})]\big]^{\mathrm{T}} P^e[t_k|t^e(t_k)]^{-1} \\ &\qquad \cdot\big\{\hat{x}^e[t_k|t^e(t_k)] - g\big[t_k, \hat{x}[t_k|t^e(t_{k-1})]\big] \\ &\qquad + G\big[t_k, \hat{x}[t_k|t^e(t_{k-1})]\big]\hat{x}[t_k|t^e(t_{k-1})]\big\}, \end{aligned} \tag{34}$$

where $t^r(t_k), t^e(t_k)$ and $t^r(t_{k-1}), t^e(t_{k-1})$ are the times of the most recent update of LT r and e prior to t_k and t_{k-1}, respectively. The approximation[5] in (39) is needed to evaluate the Jacobian matrix since $\hat{x}^E$ is not available.

The terms in the braces in (32) represent the accumulated new information from sensors r and e during the fusion window $(t_{k-1}, t_k]$ and are mapped directly to the fusion time t_k. Note that if the most recent update of LT r or LT e prior to t_k occurs prior to t_{k-1}, i.e., there is no new information from this LT during the window $(t_{k-1}, t_k]$, then the terms in the braces corresponding to each LT will be equal and thus cancel—the "new information" from this LT during this window is zero in this case.

The corresponding information matrix fusion equation is

$$\begin{aligned} &P(t_k|t_k)^{-1} \\ &\quad = P(t_k|t_{k-1})^{-1} + \{P^r[t_k|t^r(t_k)]^{-1} - P^r[t_k|t^r(t_{k-1})]^{-1}\} \\ &\qquad +\{G\big[t_k, \hat{x}^E[t_k|t^e(t_{k-1})]\big]^{\mathrm{T}} P^e[t_k|t^e(t_k)]^{-1} \\ &\qquad \cdot G\big[t_k, \hat{x}^E[t_k|t^e(t_{k-1})]\big]\} \end{aligned} \tag{35}$$

$$\begin{aligned} &\approx P(t_k|t_{k-1})^{-1} + \{P^r[t_k|t^r(t_k)]^{-1} - P^r[t_k|t^r(t_{k-1})]^{-1}\} \\ &\qquad +\{G\big[t_k, \hat{x}[t_k|t^e(t_{k-1})]\big]^{\mathrm{T}} P^e[t_k|t^e(t_k)]^{-1} \\ &\qquad \cdot G\big[t_k, \hat{x}[t_k|t^e(t_{k-1})]\big]\}. \end{aligned} \tag{36}$$

The "new information" terms in the braces in (36) are not uncorrelated from the past information even in the linear case—the uncorrelatedness holds only for full-rate communication. Their use for "decorrelation" from the past is only approximate.

The fusion architecture for an LT-driven asynchronous heterogeneous IMF is shown in Fig. 5, where the dashed circle indicates the mapping of the new information from angle space to Cartesian (39).

VI. SIMULATION RESULTS

The asynchronous heterogeneous IMF is evaluated for two cases for the scenario detailed in the sequel: (i) full-rate (LT-driven) asynchronous LTs and (ii) reduced-rate (FC-driven) asynchronous LTs. The performance of synchronous and heterogeneous LTs is also evaluated. The homogeneous and synchronous heterogeneous cases are discussed in [12] and will not be duplicated here.

A. The State Models for the Active and Passive Sensors

In the ξ–η space, a radar located at $[\xi^r\ \eta^r]$ with, for simplicity, direct Cartesian position measurements with measurement noises w^r [6] and an EO sensor located at $[\xi^e\ \eta^e]$ with bearing measurements only,

$$\theta^e = \tan^{-1}[(\eta - \eta^e)/(\xi - \xi^e)] + w^e, \tag{37}$$

are considered for the IMF for a two-dimensional (2-D) target. The measurement noises w^r and w^e are assumed to be independent zero-mean white Gaussians with corresponding standard deviations σ^r and σ^e.

The active sensor (radar) provides 2-D measurements in 2-D Cartesian space (position) and a 4-D LT state (position and velocity) with a discretized continuous time white noise acceleration (CWNA) motion

[5] $G\big[t_k, \hat{x}^E[t_k|t^e(t_{k-1})]\big] \approx G\big[t_k, \hat{x}[t_k|t^e(t_{k-1})]\big]$, $\hat{x}^E[t_k|t^e(t_{k-1})] \approx \hat{x}[t_k|t^e(t_{k-1})]$.

[6] The radar's measurements in polar coordinates can be transformed into Cartesian coordinates with an unbiased consistent transformation [3, Ch. 10.4.3].

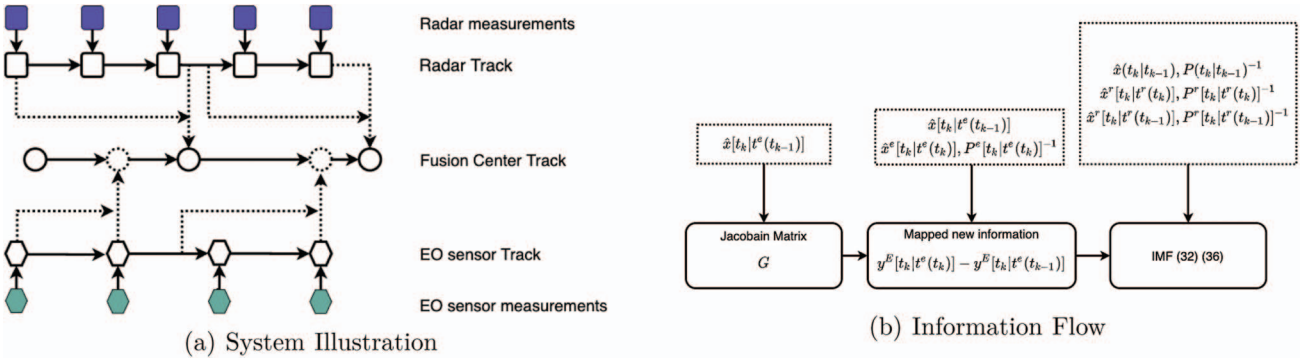


Fig. 5. Fusion architecture for FC-driven asynchronous heterogeneous IMF.

model [3] in Cartesian coordinates:

$$\mathbf{x}^r = [\xi \;\; \dot{\xi} \;\; \eta \;\; \dot{\eta}]^{\mathrm{T}} \tag{38}$$

with the discretized dynamic model to be

$$\mathbf{x}^r(t^r_{l+1}) = F^r \mathbf{x}^r(t^r_l) + \mathbf{v}^r(t^r_l), \tag{39}$$

and measurement model

$$\mathbf{z}^r = H^r \mathbf{x}^r(t^r_l) + \mathbf{w}^r(t^r_l), \tag{40}$$

where

$$F^r = \begin{bmatrix} 1 & T^r & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T^r \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{41}$$

$$H^r = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{42}$$

The process noise vector has covariance matrix

$$Q^r = \begin{bmatrix} \frac{1}{3}(T^r)^3 & \frac{1}{2}(T^r)^2 & 0 & 0 \\ \frac{1}{2}(T^r)^2 & T^r & 0 & 0 \\ 0 & 0 & \frac{1}{3}(T^r)^3 & \frac{1}{2}(T^r)^2 \\ 0 & 0 & \frac{1}{2}(T^r)^2 & T^r \end{bmatrix} \tilde{q}, \tag{43}$$

where $\tilde{q}$ is the power spectral density and $\tilde{q} = 3.8\ \mathrm{m}^2/\mathrm{s}^3$ in simulations.

The EO sensor uses a KF also based on a CWNA model with a state vector involving the angle and angle rate

$$\mathbf{x}^e = [\theta \;\; \dot{\theta}]^{\mathrm{T}}. \tag{44}$$

The discretized dynamic model is

$$\mathbf{x}^e(t^e_m + 1) = F^e \mathbf{x}^e(t^e_m) + \mathbf{v}^e(t^e_m), \tag{45}$$

$$\mathbf{z}^e = H^e \mathbf{x}^e(t^e_m) + \mathbf{w}^e(t^e_m), \tag{46}$$

where

$$F^e = \begin{bmatrix} 1 & T^e \\ 0 & 1 \end{bmatrix}, \tag{47}$$

$$H^e = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{48}$$

The state vector (38) and the state vector (44) have a nonlinear relationship

$$\mathbf{x}^e = \alpha\,[\mathbf{x}^r] \tag{49}$$

with explicit expressions

$$\theta = \operatorname{atan}\frac{\eta - \eta^e}{\xi - \xi^e}, \tag{50}$$

$$\dot{\theta} = \frac{v \sin(\phi)}{r^e}, \tag{51}$$

where v is the target speed given by

$$v = \sqrt{\dot{\xi}^2 + \dot{\eta}^2}, \tag{52}$$

r^e is the range with respect to the passive sensor's location given by

$$r^e = \sqrt{(\xi - \xi^e)^2 + (\eta - \eta^e)^2}, \tag{53}$$

and ϕ is the difference between velocity angle and position azimuth angle given by

$$\phi = \operatorname{atan}\frac{\dot{\eta}}{\dot{\xi}} - \operatorname{atan}\frac{\eta - \eta^e}{\xi - \xi^e}. \tag{54}$$

The process noise covariance matrix of the EO tracker's model at time t_k has the following relationship

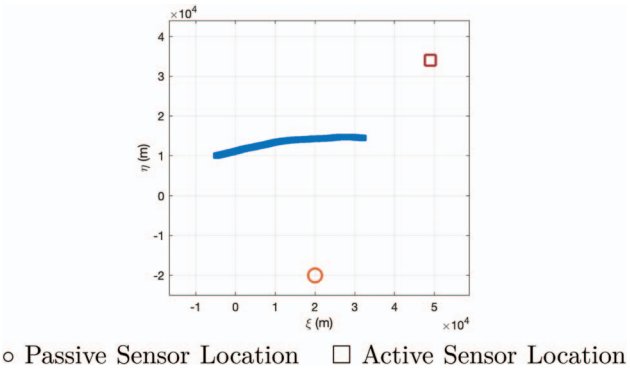


Fig. 6. Target trajectory (one realization) and sensor locations.

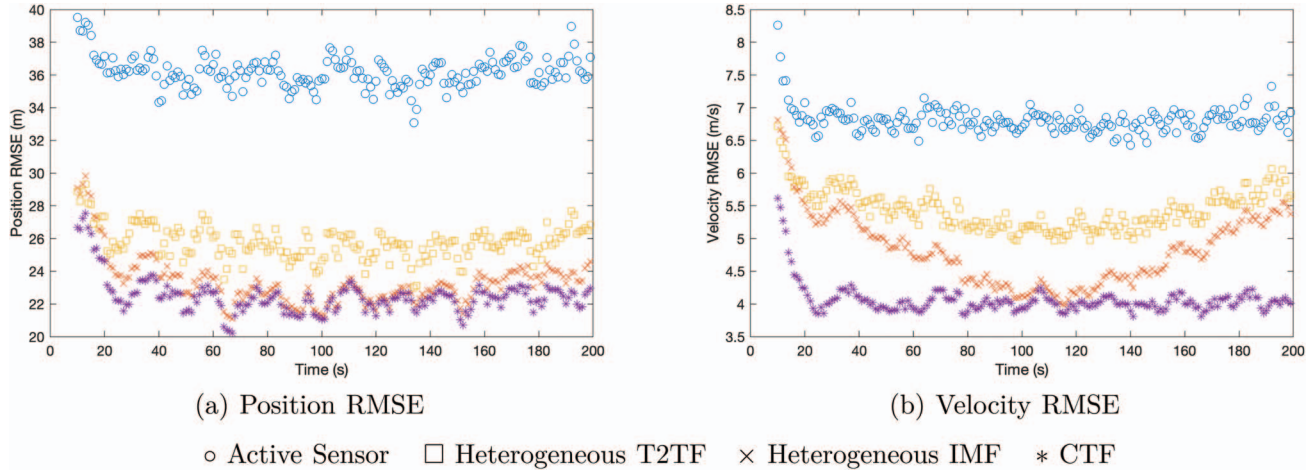


Fig. 7. Full-rate (LT-driven) heterogeneous IMF RMSE from 500 runs.

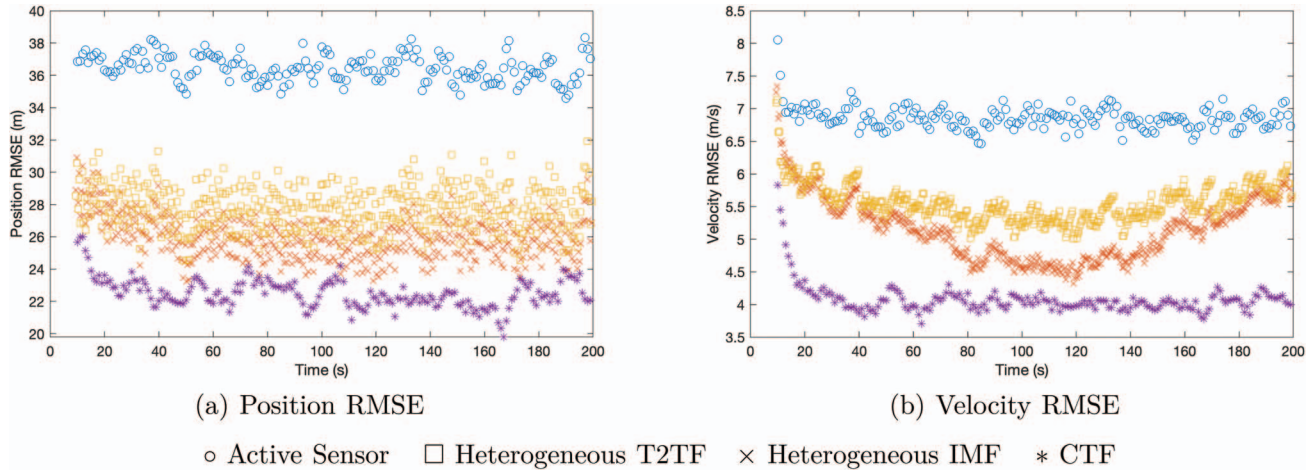


Fig. 8. Reduced-rate ($T^{\mathrm{FC}} = 0.4$ s) heterogeneous IMF RMSE from 500 runs.

with the active process noise covariance matrix, as derived and discussed in [12][7]:

$$Q^e(t_{k+1}, t_k) = A(t_k) Q^r(t_{k+1}, t_k) A(t_k)', \tag{55}$$

where

$$A(t_k) \triangleq \left[\nabla_x \alpha(x)^{\mathrm{T}}\right]^{\mathrm{T}}\Big|_{x=F^r[t_{k+1},t_k]\mathbf{x}^r(t_k)}. \tag{56}$$

B. Numerical Results

The sensor locations are [49 34] km and [−20 20] km for the active sensor and passive sensor, respectively. The target is assumed to have an initial position [−5 10] km and velocity [200 20] m/s. The trajectory lasts for 200 s. Fig. 6 shows the target trajectory (one realization) and sensor locations. The standard deviations of measurement noises are assumed to be $\sigma^r = 50$ m for the active sensor (direct position measurement in both coordinates) and $\sigma^e = 0.4$ milliradian (mrad) for the passive sensor (azimuth angle). In all asynchronous cases, the active sensor (radar) has sampling interval $T^r = 1$ s and the passive sensor (EO) has sampling interval $T^e = 0.1$ s.

Several FC sampling intervals (fusion rates) are used in the simulation to compare the performance of the proposed algorithm. The simulation results are based on 500 Monte Carlo runs. To evaluate the performance of the IMF: (i) the full-rate centralized tracking/fusion is carried out, which is the optimal one can achieve and (2) the heterogeneous T2TF [13] is also carried out. Note that the RMSE results in Figs 7–10 started at 9 s after the convergence of LFs to avoid large plot scales.

The reduced-rate asynchronous heterogeneous IMF is evaluated with multiple sampling rates at the FC: $T_{\mathrm{FC}} = 0.4, 0.8$, and 1.6 s. The RMSEs for both position and velocity are evaluated. Fig. 7 shows the RMSE of the full-rate IMF. In this case, the rate is the that of a higher rate sensor (10 Hz, since $T^e = 0.1$ s). Simulation results for $T_{\mathrm{FC}} = 0.4, 0.8$, and 1.6 s are shown in Figs. 8, 9, and 10, respectively. The oscillations of the position errors of the IMF are due to the fact that at its update time it uses a predicted active LT (radar) state since the FC is not synchronized with the radar. It can be seen that with full-rate communication, the proposed

[7]This process noise covariance mapping is similar to [8] except for the linearization Jacobian, which is evaluated at the IMF fused state, while [8] used a "worst-case"-based mapping.

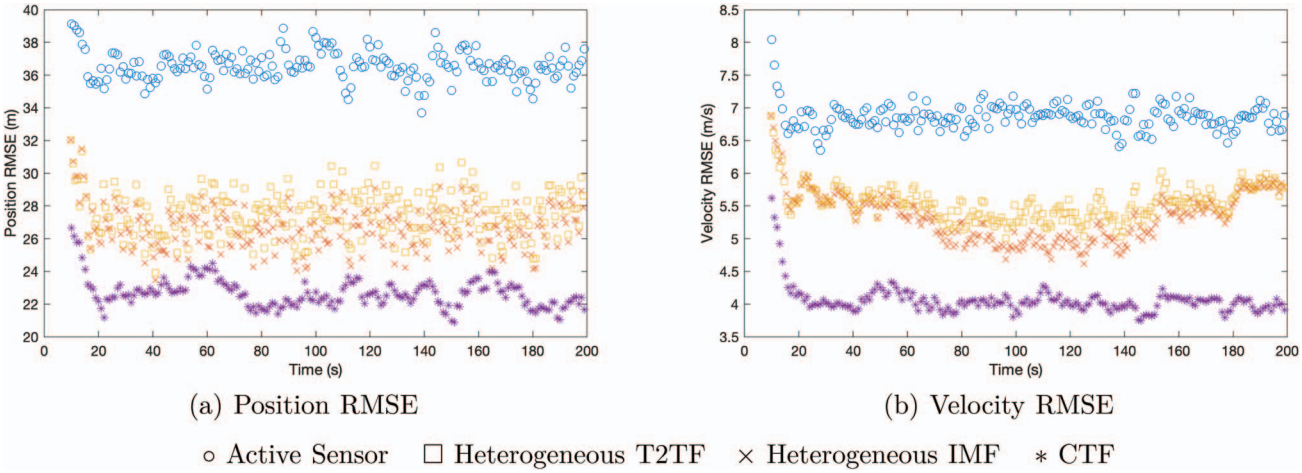


Fig. 9. Reduced-rate ($T^{\mathrm{FC}} = 0.8$ s) heterogeneous IMF RMSE from 500 runs.

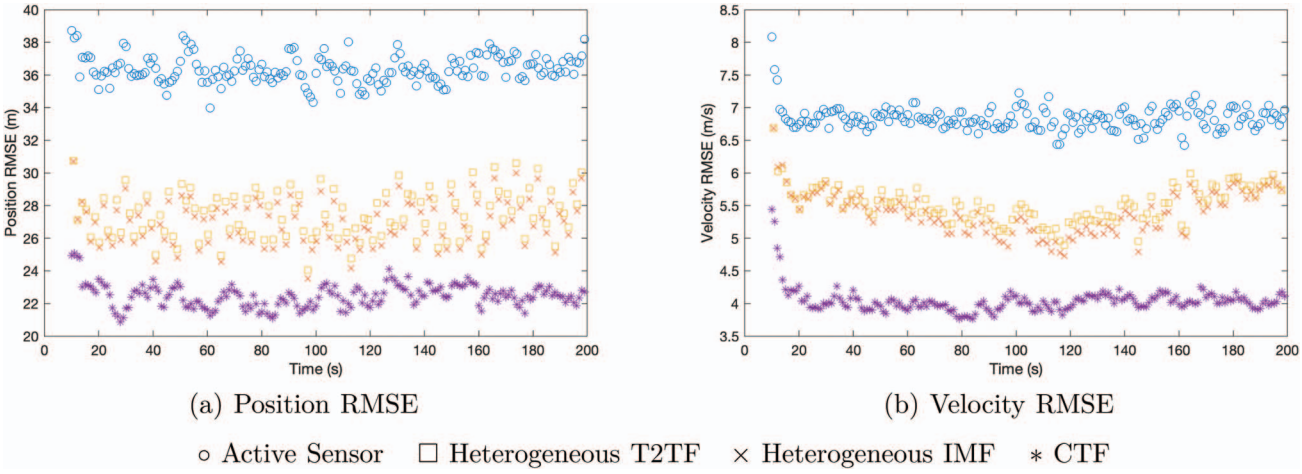


Fig. 10. Reduced-rate ($T^{\mathrm{FC}} = 1.6$ s) heterogeneous IMF RMSE from 500 runs.

heterogeneous and asynchronous IMF achieves almost the optimal result (the CTF result) for position. The performance of the velocity fusion is somewhat off due to the nonlinearity (linear motion in Cartesian space is not linear in angle space), and approximation of the "new information," The results also depend on the geometry between the sensors and the target trajectory. It can be seen that a larger sampling interval at the FC will degrade the performance of the IMF; however, there is always a reduction in the RMSE compared to the case with an active sensor only for both position and velocity. In all the cases considered, the proposed IMF has better performance than T2TF by having a smaller RMSE for both position and velocity.

VII. CONCLUSION

In this work, the IMF algorithm was extended to nonlinear, asynchronous, and heterogeneous systems. The LTs from an active sensor and a passive sensor are in different state spaces and are related by a nonlinear transformation without inverse. Both the LT-driven full-rate asynchronous case and FC-driven reduced-rate asynchronous case are investigated. Although the passive (EO/IR) LT state with a lower dimension cannot be used directly in the IMF, it has been shown that its new information can be mapped to the high-dimension state space and then used by the IMF at the FC. With full-rate communication (LT driven), the proposed IMF can almost achieve the optimal solution (full-rate CTF). The performance of the FC-driven asynchronous IMF is not optimal but still remarkable compared with the results from the active sensor (radar) only and T2TF by achieving a smaller RMSE in both position and velocity. Real data testing is not available at the current stage; however, it will be investigated in future works.

REFERENCES

[1] C. Allig and G. Wanielik
"Heterogeneous track-to-track fusion using equivalent measurement and unscented transform," in
Proc. 21st Int. Conf. Inf. Fusion, Cambridge, U.K., Jul. 2018.

[2] M. Aeberhard, S. Schlichtharle, N. Kaempchen, and T. Bertram
"Track-to-track fusion with asynchronous sensors using information matrix fusion for surround environment perception,"

IEEE Trans. Intell. Transp. Sys., vol. 13, no. 4, pp. 1717–1726, Dec. 2012.

[3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation With Applications to Tracking and Navigation: Theory, Algorithms and Software. New York, NY, USA: Wiley, 2001.

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

[5] K. C. Chang, C.-Y. Chong, and S. Mori
"Analytical and computational evaluation of scalable distributed fusion algorithms,"
IEEE Trans. Aerosp. Electron. Syst., vol. 46, no. 4, pp. 2022–2034, Oct. 2010.

[6] C. Y. Chong
"Hierarchical estimation,"
in *Proc. MIT/ONR Workshop C3*, Monterey, CA, USA, 1979.

[7] M. Mallick, K. C. Chang, S. Arulampalam, and Y. Yan
"Heterogeneous track-to-track fusion in 3D using IRST sensor and air MTI radar,"
IEEE Trans. Aerosp. Electron. Syst., vol. 55, no. 6, pp. 3062–3079, Dec. 2019.

[8] T. L. Song, H. W. Kim, and D. Musicki
"Distributed (nonlinear) target tracking in clutter,"
IEEE Trans. Aerosp. Electron. Syst., vol. 51, no. 1, pp. 654–668, Apr. 2015.

[9] X. Tian and Y. Bar-Shalom
"Algorithms for asynchronous track-to-track fusion,"
J. Adv. Inf. Fusion, vol. 5, no. 2, pp. 128–138, Dec. 2010.

[10] X. Tian, Y. Bar-Shalom, T. Yuan, E. Blasch, K. Pham, and G. Chen
"A generalized information matrix fusion based heterogeneous track-to-track fusion algorithm,"
in *Proc. SPIE Conf. Signal Processing, Sensor Fusion, Target Recognition*, Orlando, FL, USA, Apr. 2011.

[11] K. Yang, Y. Bar-Shalom, and P. Willett
"Asynchronous and heterogeneous track-to-track fusion with cross-covariance,"
J. Adv. Inf. Fusion, vol. 15, no. 1, pp. 39–48, June 2020.

[12] K. Yang, Y. Bar-Shalom, and K. C. Chang
"Information matrix fusion for nonlinear, asynchronous and heterogeneous systems,"
in *Proc. 22nd Int. Conf. Inf. Fusion*, Ottawa, Canada, Jul. 2019.

[13] T. Yuan, Y. Bar-Shalom, and X. Tian
"Heterogeneous track-to-track fusion,"
J. Adv. Inf. Fusion, vol. 6, no. 2, pp. 131–149, Dec. 2011.

Kaipei Yang received the B.S. degree from Northwestern Polytechnical University in 2014 and the Ph.D. degree from the University of Connecticut in 2019. She is now an Assistant Research Professor with the Department of Electrical and Computer Engineering, University of Connecticut. Her research interests include statistical signal processing, estimation theory, and information fusion. She gained experience in autonomous driving vehicles while working at NIO in San Jose, CA, USA, in 2018.

Yaakov Bar-Shalom received the B.S. and M.S. degrees in electrical engineering from the Technion in 1963 and 1967, respectively, and the Ph.D. degree in electrical engineering from Princeton University in 1970. From 1970 to 1976, he was with the Systems Control, Inc., Palo Alto, CA, USA. Currently, he is a Board of Trustees Distinguished Professor with the Department of Electrical and Computer Engineering and a Marianne E. Klewin Professor in Engineering with the University of Connecticut. Since 1995, he has been a Distinguished Lecturer with the IEEE Aerospace and Electronic Systems Society. His current research interests are in estimation theory, target tracking, and data fusion. He has published more than 650 papers and book chapters. He has coauthored/edited eight books, including *Tracking and Data Fusion* (YBS Publishing, 2011). He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He served as an Associate Editor for the IEEE Transactions on Automatic Control and *Automatica*. He was General Chairman of the 1985 ACC. He served as the 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 2004–2013. In 1987, he received the IEEE CSS Distinguished Member Award. He is a co-recipient 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, he was awarded 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 was renamed in 2016 as the Yaakov Bar-Shalom Award for a Lifetime of Excellence in Information Fusion.

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 with the Advanced Decision Systems (ADS) Division, Booz-Allen & Hamilton, Mountain View, CA, USA. 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, multisensor data fusion, Bayesian inference, machine learning, and financial engineering. He has more than 35 years of industrial and academic experience and published more than 200 papers in the areas of multitarget tracking, distributed sensor fusion, Bayesian network 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 for the IEEE Transactions on Systems, Man, and Cybernetics, from 2002 to 2007. He was the Technical Program Co-Chair of the 2009 International Conference on Information Fusion, Seattle, CA, USA. Dr. Chang was elected as a Fellow of IEEE for his contribution on "sensor data fusion and Bayesian probabilistic inference" in 2010.

Track-to-Track Fusion Using Inside Information From Local IMM Estimators

RADU VISINA
YAAKOV BAR-SHALOM
PETER WILLETT
DIPAK K. DEY

A novel approach to the track-to-track fusion (T2TF) of state estimates from interacting multiple-model (IMM) estimators using inside information [mode-conditioned estimates (MCEs) and mode probabilities] is described in this paper. Fusion is performed on-demand, i.e., without conditioning on past track data. The local trackers run IMM estimators to track a maneuvering target with switching process noise and they transmit MCEs and mode probabilities to a fusion center. The fused state posterior probability density is a Gaussian mixture, where the parameters of the required likelihood functions can be computed recursively. Mode probabilities are fused by transforming them to log-ratios and using them as statistical information in the likelihood function of the mode. This results in consistent data fusion based on known target and local tracker (IMM) parameters. Simulations show that this method outperforms the fusion of the local IMM estimator Gaussian-approximated outputs both in terms of error during target maneuvers and in terms of the consistency of the mean-squared error (MSE). It is a generalization of Gaussian T2TF with crosscovariance, and its performance is close to that of centralized measurement fusion (CMF)—by accounting for the error and log-ratio crosscovariances, the fused covariance consistency matches the ideal consistency of CMF without requiring memory of past fused tracks. The method is also shown to be more accurate, informative, consistent in MSE, and of lower computational and communication cost than Chernoff fusion, a recently published method for Gaussian mixture fusion.

Manuscript received May 7, 2019; revised December 11, 2019 and June 10, 2020; released for publication July 27, 2020.

This work was supported in part by NSWC under Grant N001741810004.

Refereeing of this contribution was handled by Chee-Yee Chong.

The authors are with the University of Connecticut, Storrs, CT 06269-4157 USA (E-mail: radu.visina, yaakov.bar-shalom, peter.willett, dipak.dey@uconn.edu).

1557-6418/20/$17.00

I. INTRODUCTION

The interacting multiple model (IMM) estimator is a powerful nonlinear state estimator for targets whose dynamic evolution model changes according to a discrete-time, discrete-state Markov chain with known transition probabilities, and it may be used in local estimators for tracking maneuvering targets or other mode-switching systems. In this work, the posterior probability density function (PDF) of the state of a dynamic target, conditioned on information from local trackers (LTs) implementing the IMM estimation algorithm [4], is derived for on-demand track-to-track fusion (T2TF). The LTs provide Gaussian mixture track information from inside their IMM algorithms [the current mode-conditioned estimates (MCEs) and mode probabilities]. The fusion center (FC) continuously updates a linearized system description of the IMM estimator's error and mode probability behavior to compute the required parameters of the likelihood functions of the state and mode. The fused posterior state PDF is then approximated as a Gaussian mixture.

When new measurements from every sensor can be communicated to a FC at every measurement time, the optimal solution is to stack all new measurements into a single vector and run a single estimator, resulting in optimal centralized measurement fusion (CMF) [5]. However, data may need to be sent at arbitrarily low rates compared to the LT measurement intervals, requiring the transmission of recursively computed local estimates (and sometimes covariances). The problem is difficult because of the dependent nature of the received state estimation errors. The correlation between the local estimation errors was described in [2] and [3] as the recursively-computed crosscovariance matrices for linear, Gaussian estimators, and their incorporation into the standard fusion equations results in optimal fusion (given only the on-demand tracks) and consistent fused covariances. This method is termed Gaussian T2TF with crosscovariance (GT2TFwXC) and requires knowledge of the Kalman filter design parameters.

The recursive computations described in this paper yield the required matrices (including crosstracker and crossmode covariances) for IMM track fusion as a multiple-model generalization of GT2TFwXC. For a single mode, the algorithm reduces naturally to GT2TFwXC. Just as GT2TFwXC requires knowledge of the LT Kalman filter parameters and the target, this paper's proposed method also requires the parameters of the LT IMM estimators and the target. To bound the complexity of the problem, the proposed method is derived for trackers that agree on the set of possible dynamic target maneuvering modes and the mode transition probabilities. The information from the trackers is for the same times (i.e., it is synchronous). It is also assumed that the target state transition matrix is the same in both modes, so the method is ideal for target models that switch process noise covariance only.

(Section III-E explains the difficulty encountered with a switching transition matrix.) The assumptions stated here serve to introduce important theory from which additional complexities can be included in future work. The proposed method, along with the theory supporting it, are the foundations for on-demand T2TF from multiple-model trackers.

For the fusion of IMM mode-conditioned information, the extraction of track information from mode probabilities (i.e., the mixture weights) is an additional problem. This suggests that the received probabilities should be treated as statistical information in the conditioning of the posterior (fused) PDF. To account for the dependency between the received probabilities, they are transformed into infinite-support *log-ratios (LRs) of probabilities* and a linear approximation of their evolution is derived. Using this technique, the local mode probabilities are successfully combined to form the fused mode probabilities with the same number of modes, allowing for mode inference based on the fused information.

As an alternative, T2TF can be performed naively using the LT's moment-matched IMM output estimate and covariance [11], but that method has poor performance during maneuvers and does not account for error cross-covariance.

An alternate Gaussian mixture fusion approach has been explored—Chernoff fusion, first proposed by Mahler [13] and Hurley [10], has received some attention in the literature, and is capable of minimizing the fused mean-squared error (MSE) while assuring that the fused covariance is greater than or equal to the actual sample MSE, without direct knowledge of the error crosscovariances. A successful, computationally feasible implementation for Gaussian mixtures using unscented sigma points has been developed in [8] and used in distributed fusion from IMM tracks in [9]. However, Chernoff fusion is unable to exploit system model information (i.e., target dynamic motion models and IMM design parameters, assumed available in this paper) and may produce fused covariance values that are too high, though they acceptably represent the sample MSE of the fused estimate (i.e., the covariances are consistent). The sigma point implementation is still computationally demanding due to the need to search for the optimal fusion exponent, and requires the transmission of local mode-conditioned estimate covariances. Just as the IMM fusion proposed here generalizes GT2TFwXC, Chernoff fusion generalizes the covariance intersection method and solves the fusion problem when the crosscovariance cannot be computed or because system parameters (local IMM parameters) are not available. While ignoring crosscovariances altogether results in overly optimistic fused covariances, Chernoff Fusion results in conservative fused covariances, which are still not ideal, meaning that the MSE is higher than what is possible with a more optimal method. When the target and LT system design parameters are known, the crosscovariances can be computed recursively and Bayesian fusion can be performed as shown in this paper without the transmission of the local covariance matrices and without the need for numerical optimization. The simulation results show that although the fused covariance of the Chernoff method match the MSE of the fused estimate, the model-driven fusion with crosscovariance presented here significantly outperforms Chernoff fusion in terms of fused accuracy. Another advantage of the method in this paper is that the fused probability density output includes fused mixture probabilities, which directly provide inference about target dynamic maneuvering mode. Chernoff fusion cannot provide this output information because the number of mixture components in its fused PDF is a product of the number of local components, and such a mixture has no event-based interpretation in the multiple-model target maneuvering scenario.

Another approach to the fusion of Gaussian mixture filter outputs was developed in [14]. That paper introduces the topic of crosscovariances for every mixture component. However, that approach mainly treats problems with Gaussian mixture process noise and Gaussian mixture measurement PDFs, both of which are unlike the Markov chain switching processes involved in maneuvering target tracking. Additionally, there is no consideration of mixture component reduction strategies that complicate the crosscovariance structure (such as the mixing process of the IMM estimator); they assume fully invertible state-to-measurement equations (unrealistic in target tracking applications where the state vector is longer than the measurement vector), and do not consider the dependent, stochastic nature of local mixture probabilities (weights) in their fused mixture probabilities. The method developed in this paper accounts for the Markov chain process of the target maneuvers and the mixing process of the IMM, does not require invertible measurement equations, and fully computes the crosscovariances between all local mode-conditioned state estimates and mode probabilities at the FC.

Though the present method does not require memory of past fused tracks, alternative data fusion schemes exist that utilize memory of past fused tracks and decorrelate the information being passed throughout a distributed sensing network. One of the first such algorithms for Gaussian tracks was Information Matrix Fusion [5]. See [6] and [7] for a general discussion of recent advancements in distributed tracking. An approximate method to solve this for IMM tracks was developed in [12]. A model-agnostic method for fusion of IMM Gaussian mixture tracks with memory was also developed in [1]. These methods do not provide the crosscovariance for fusion without past track information. Given these distinctions, further comparison is outside of the scope of this paper; however, it should be noted that, theoretically fusion schemes with memory running at full rate could yield the accuracy performance of CMF (which is slightly more accurate than on-demand fusion techniques [5]), but the results in this paper show

that the present memoryless method still achieves fused track accuracy close to that of CMF and with ideal MSE consistency.

Even when using a suitable reduced-rate T2TF method, the problem of initial fusion must still be solved using an on-demand fusion scheme. Without it, the FC has no initial condition (i.e., prior) for recursive updating of the fused track. This resembles the problem of running a Kalman Filter without a previous estimate and covariance [4], [5]. GT2TFwXC provides a consistent fused estimate without initial conditions for linear Gaussian systems; likewise, this paper shows how this is accomplished when the received tracks come from IMM estimators tracking a maneuvering target.

This paper is organized as follows: Section II introduces the problem mathematically, Section III describes the required Bayesian fusion theory, Section IV develops the algorithmic steps required to implement fusion with IMM inside information, Section V summarizes the algorithm and discusses computational complexity, and Section VI presents Monte Carlo simulations and results. A list of symbols and acronyms is provided in Table 1 for reference.

II. DESCRIPTION OF TARGET AND LOCAL TRACKERS

For clarity, only two LTs and two dynamic modes will be considered, but the extension to multiple trackers and modes is possible. Local state estimation is performed by two trackers obtaining noisy observations of a target whose dynamics may switch between two different modes. Each tracker, indexed $j = 1, 2$, computes MCEs and mode-conditioned covariances (MCCs) of $\mathbf{x}(k)$ from modes indexed $m = 1, 2$. With $\mathbf{Z}_j^k$ as the vector of all measurements at tracker j, up to and including the present time step,

$$\mathbf{Z}_j^k = \left[\mathbf{z}_j(0)'\ \mathbf{z}_j(1)'\ \ldots\ \mathbf{z}_j(k)'\right]', \tag{1}$$

the N_x-dimensional MCEs, conditioned on the current target mode $M(k)$ being m, are denoted and defined as

$$\hat{\mathbf{x}}_j^m(k|k) \triangleq E[\mathbf{x}(k) \mid \mathbf{Z}_j^k, M(k) = m], \quad \begin{matrix} j = 1, 2, \\ m = 1, 2 \end{matrix} \tag{2}$$

and with the MCE errors (MCEEs) defined as

$$\tilde{\mathbf{x}}_j^m(k|k) = \hat{\mathbf{x}}_j^m(k|k) - \mathbf{x}(k), \tag{3}$$

the MCCs are

$$\mathbf{P}_j^m(k|k) \triangleq E\left[\tilde{\mathbf{x}}_j^m(k|k)\tilde{\mathbf{x}}_j^m(k|k)' \mid \mathbf{Z}_j^k, M(k) = m\right]. \tag{4}$$

The true state of the target, when in mode $M(k) = n$, evolves linearly in time as[1]

$$\mathbf{x}^{|n}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{v}^n(k). \tag{5}$$

The N_z-dimensional measurements of the target at each tracker are obtained according to

$$\mathbf{z}_j(k) = \mathbf{H}_j\mathbf{x}(k) + \mathbf{w}_j(k). \tag{6}$$

The trackers compute the probability of the target being in mode m at time step k as

$$\mu_j^m(k) \triangleq \mathrm{P}\left(M(k) = m \mid \mathbf{Z}_j^k\right). \tag{7}$$

The evolution of the target's dynamic modes is modeled as a Markov chain. Its known transition probability matrix (TPM; [4]) is

$$\Pi = \begin{bmatrix} \pi^{11} & \pi^{12} \\ \pi^{21} & \pi^{22} \end{bmatrix} \tag{8}$$

and all of its rows must have a sum of 1.

Two key simplifying assumptions are made. First, as can be seen in equation (5), mode changes affect the process noise only and not the state transition matrix $\mathbf{F}$. This simplification was made because the theory required to fuse with switching $\mathbf{F}$ becomes much more involved, especially if the state space of the models switches dimension. It is important to note that the methods developed in this paper are the foundations of a unique methodology in on-demand T2TF that can be extended to more complexities in future work (see Section III-E). The Bayesian derivation of Gaussian mixture posteriors, along with obtaining the parameters of the likelihood function(s) through *linearization and recursion* of the joint system describing the trackers and the target, constitutes the powerful, yet fundamental, stochastic systems approach to T2TF proposed in this paper.

The second simplification in this paper is that mode switching does not affect the measurement equation (6). Mode-specific measurement parameters can be substituted if required, as long as they switch as part of the same Markov chain process of the target. Given the independence of the target motion and measurement system(s), such examples are not typical in tracking scenarios, so treatment of this case is beyond the scope of this paper.

III. BAYESIAN THEORY FOR IMM INSIDE INFORMATION FUSION

A. The Posterior Fused State PDF

Omitting the time-step index k for brevity, the posterior state PDF of the target state using the data from

[1]Note that m is the index of the MCE at the LT. The FC must consider the received MCE and probabilities under all mode hypotheses, so n is used in the multiple-model inference process at the FC, while m is used only to index the received data.

Table 1
List of Symbols and Acronyms

$\bar{(\cdot)}$	Mean of $(\cdot)$
$\breve{(\cdot)}$	Mixed initial condition of $(\cdot)$ (IMM algorithm)
$\hat{(\cdot)}$	Estimate of $(\cdot)$
$\tilde{(\cdot)}$	Error of $\hat{(\cdot)}$
$\mathbf{d}_j^{\vert n}(k)$	Gaussian-approximated process noise entering the LR
$\mathbf{D}_j^{\vert n}(k)$	Covariance of $\mathbf{d}_j(k)$
$\mathbf{F}$	State transition matrix
$\mathbf{g}^{\vert n}(k)$	Additive noise of the linearized joint IMM system
$\mathbf{G}^{\vert n}(k)$	Covariance of $\mathbf{g}(k)$
$\mathbf{H}_j$	Measurement matrix at LT j
$\mathbf{I}_{N\times N}$	$N \times N$ identity matrix
j	LT index ($j = 1, 2$ — used as a subscript)
$\mathbf{J}^{\vert n}(k)$	Transition Jacobian matrix of the joint IMM system
k	Discrete time step
LR	Log-ratio(s) of probability pair(s)
LT	Local tracker(s)
m	Target mode index of the received estimates and probabilities ($m = 1, 2$ — used as a superscript)
MCC	Local mode-conditioned estimate covariance
MCE(E)	Local mode-conditioned estimate (error)
MCI	Local mode-conditioned innovation (i.e. residual)
MCP(E)	Local mode-conditioned prediction (error)
$M(k)$	True target dynamic mode
$\mu_j^m(k)$	Locally-computed probability of mode m
$\mu_j(k)$	$[\mu_j^1(k) \quad \mu_j^2(k)]'$
$\mu_j^{lm}(k)$	Initial condition mixing weight (IMM algorithm)
n	Target mode index hypothesis under consideration at the FC ($n = 1, 2$)
$(\cdot)^{\vert n}$	Any variable $(\cdot)$ derived under the hypothesis that the current mode is n
$\nu_j^m(k)$	MCI
$\mathbf{N}_j^{lm}$	Spread-of-the-means term of a mixture's covariance
$\omega_j(k)$	LR of the mode based on $\mu_j^1(k\vert k)$
$\mathbf{\Omega}_j^{\vert n}(k)$	Covariance of $[\omega_1(k)\ \omega_2(k)]'$
$p(\cdot)$	Any probability density function (PDF)
$P(\cdot)$	Any probability mass function (PMF)
$\mathbf{P}_{\mathrm{F}}(k)$	Covariance of the moment-matched fused estimate output error
$\mathbf{P}_{\mathrm{F}}^n(k)$	Covariance of the fused nth MCEE
$\mathbf{P}_j^m(k\vert k)$	Locally-computed covariance of $\tilde{\mathbf{x}}_j^m(k\vert k)$ (MCC)
$\breve{\mathbf{P}}_j^m(k\vert k)$	Locally-computed covariance of $\breve{\mathbf{x}}_j^m(k\vert k)$
$\mathbf{P}^{\vert n}(k\vert k-1)$	FC-computed complete covariance/crosscovariance of $\tilde{\mathbf{x}}_j^m(k\vert k-1)$
$\mathbf{P}^{\vert n}(k)$	FC-computed complete covariance/crosscovariance of the MCEE
$\phi_j(k)$	Received data from LT j
π^{lm}	Markov Chain transition probability from mode l to mode m
$\mathbf{Q}^n$	Covariance of the process noise under mode n
$\mathbf{R}_j$	Covariance of measurement error at LT j
$\mathbf{S}^{\vert n}(k)$	FC-computed complete covariance/crosscovariance of the MCI
$\mathbf{S}_j^m(k)$	Locally-computed covariances of $\nu_j^m(k)$
TPM	Transition probability matrix of Markov chain
$\mathbf{v}^n(k)$	Zero mean process noise under mode n
$\mathbf{w}_j(k)$	Zero mean measurement error at LT j
$\mathbf{W}_j^m(k)$	Locally computed Kalman gain matrices
$\mathbf{x}_{\mathrm{F}}(k)$	Moment-matched fused estimate output
$\hat{\mathbf{x}}_{\mathrm{F}}^{\vert n}(k)$	Fused nth MCE
$\breve{\mathbf{x}}_j^m(k\vert k)$	mth mode's mixed initial conditions (from IMM algorithm)
$\hat{\mathbf{x}}_j^m(k\vert k-1)$	Local tracker MCP
$\tilde{\mathbf{x}}_j^m(k\vert k-1)$	Local tracker MCPE
$\hat{\mathbf{x}}_j^m(k\vert k)$	Local tracker MCE
$\tilde{\mathbf{x}}_j^m(k\vert k)$	Local tracker MCEE
$\mathbf{x}(k)$	True target state
$\mathbf{y}^{\vert n}(k)$	"State vector" of the joint IMM system, computed at the FC
$\mathbf{Y}^{\vert n}(k)$	Covariance of $\mathbf{y}^{\vert n}(k)$
$\mathbf{z}_j(k)$	Measurement at LT j

two trackers is a mixture density described by

$$p(\mathbf{x} \mid \phi_1, \phi_2) = \sum_{n=1}^{2} p(\mathbf{x} \mid \phi_1, \phi_2, M = n) P(M = n \mid \phi_1, \phi_2) \tag{9}$$

with the received data from the two trackers (MCE and mode probabilities) defined as

$$\phi_j \triangleq \left\{\hat{\mathbf{x}}_j^1(k|k), \hat{\mathbf{x}}_j^2(k|k), \mu_j^1(k)\right\}, \quad j = 1, 2, \tag{10}$$

where $\mu_j^2(k)$ is ignored in equation (10) due to its redundancy.

The FC does not have access to any $\mathbf{z}_j(k)$ or any past $\hat{\mathbf{x}}_j^m(k|k)$, but should provide the best fused estimate and its error covariance when receiving the latest MCE and mode probabilities from all LTs. To do this, the posterior fused mode-conditioned densities $p(\mathbf{x} \mid \phi_1, \phi_2, M = n)$ and posterior fused mode probabilities $P(M = n \mid \phi_1, \phi_2)$ will be derived next.

The LT MCCs $\mathbf{P}_j^m(k|k)$ are not part of the data used for fusion in equation (10) (see Appendix: Property 7), although the authors are not discouraging the transmission of these data if system parameters are unavailable. The fused MSE and covariance consistency from the Monte Carlo simulations of Section VI are extremely close to that of CMF and this offers empirical evidence that the MCCs do not contain significant information about the target state, so considerable communication savings can be achieved if the covariances are not transmitted.

B. The Fused Mode-Conditioned State Estimates

We claim that, conditioned on the received MCE, the received mode probabilities do not contribute additional information about any mode-conditioned state vector. This is proved for the linearized joint system model in Section III-C and Property 3 of the Appendix. So, the fused posterior mode-conditioned PDFs from equation (9) are approximately

$$\begin{aligned} p(\mathbf{x} \mid \phi_1, \phi_2, n) &\approx p\left(\mathbf{x} \mid \hat{\mathbf{x}}_1^1, \hat{\mathbf{x}}_1^2, \hat{\mathbf{x}}_2^1, \hat{\mathbf{x}}_2^2, M = n\right) \\ &= \frac{1}{a} p(\hat{\mathbf{x}}_1^1, \hat{\mathbf{x}}_1^2, \hat{\mathbf{x}}_2^1, \hat{\mathbf{x}}_2^2 \mid M = n, \mathbf{x}) \\ &\quad \times p(\mathbf{x} \mid M = n), \quad n = 1, 2, \end{aligned} \tag{11}$$

with a a normalizing constant and $p(\mathbf{x} \mid M = n)$ considered noninformative (i.e., diffuse) because there is no initial condition about the target state at the FC (the key assumption for on-demand fusion).[2] The likelihood of the state in equation (11) is the PDF of the LT MCE, conditioned on the true state $\mathbf{x}(k)$ and true mode $M(k) = n$, given as

$$p(\hat{\mathbf{x}}_1^1, \hat{\mathbf{x}}_1^2, \hat{\mathbf{x}}_2^1, \hat{\mathbf{x}}_2^2 \mid M = n, \mathbf{x}) \tag{12}$$

with mean

$$E\left\{\left[(\hat{\mathbf{x}}_1^1)' (\hat{\mathbf{x}}_1^2)' (\hat{\mathbf{x}}_2^1)' (\hat{\mathbf{x}}_2^2)'\right]' \mid M = n, \mathbf{x}\right\} = \left[\mathbf{x}' \mathbf{x}' \mathbf{x}' \mathbf{x}'\right]' \tag{13}$$

and covariance (to be computed recursively at the FC as described in Section IV-B)

$$\mathbf{P}^{|n} \triangleq E\left\{\begin{bmatrix} \tilde{\mathbf{x}}_1^1(\tilde{\mathbf{x}}_1^1)' & \tilde{\mathbf{x}}_1^1(\tilde{\mathbf{x}}_1^2)' & \tilde{\mathbf{x}}_1^1(\tilde{\mathbf{x}}_2^1)' & \tilde{\mathbf{x}}_1^1(\tilde{\mathbf{x}}_2^2)' \\ \tilde{\mathbf{x}}_1^2(\tilde{\mathbf{x}}_1^1)' & \tilde{\mathbf{x}}_1^2(\tilde{\mathbf{x}}_1^2)' & \tilde{\mathbf{x}}_1^2(\tilde{\mathbf{x}}_2^1)' & \tilde{\mathbf{x}}_1^2(\tilde{\mathbf{x}}_2^2)' \\ \tilde{\mathbf{x}}_2^1(\tilde{\mathbf{x}}_1^1)' & \tilde{\mathbf{x}}_2^1(\tilde{\mathbf{x}}_1^2)' & \tilde{\mathbf{x}}_2^1(\tilde{\mathbf{x}}_2^1)' & \tilde{\mathbf{x}}_2^1(\tilde{\mathbf{x}}_2^2)' \\ \tilde{\mathbf{x}}_2^2(\tilde{\mathbf{x}}_1^1)' & \tilde{\mathbf{x}}_2^2(\tilde{\mathbf{x}}_1^2)' & \tilde{\mathbf{x}}_2^2(\tilde{\mathbf{x}}_2^1)' & \tilde{\mathbf{x}}_2^2(\tilde{\mathbf{x}}_2^2)' \end{bmatrix} \middle| n, \mathbf{x}\right\}. \tag{14}$$

With equation (14) computed, the solution to equation (11) is the standard linear minimum mean-square error (LMMSE) fusion given by (see [5])

$$\hat{\mathbf{x}}_{\mathrm{F}}^n(k) = \left[\bar{\mathbf{L}}'(\mathbf{P}^{|n})^{-1}\bar{\mathbf{L}}\right]^{-1} \bar{\mathbf{L}}'(\mathbf{P}^{|n})^{-1}\hat{\mathbf{X}} \tag{15}$$

and the corresponding fused covariance given by

$$\mathbf{P}_{\mathrm{F}}^n(k) = \left[\bar{\mathbf{L}}'(\mathbf{P}^{|n})^{-1}\bar{\mathbf{L}}\right]^{-1} \tag{16}$$

with

$$\bar{\mathbf{L}} \triangleq \left[\mathbf{I}_{N_x \times N_x}\ \mathbf{I}_{N_x \times N_x}\ \mathbf{I}_{N_x \times N_x}\ \mathbf{I}_{N_x \times N_x}\right]'. \tag{17}$$

The $4N_x$-element vector $\hat{\mathbf{X}}$ in equation (15) is

$$\hat{\mathbf{X}} = \left[(\hat{\mathbf{x}}_1^1)' (\hat{\mathbf{x}}_1^2)' (\hat{\mathbf{x}}_2^1)' (\hat{\mathbf{x}}_2^2)'\right]'. \tag{18}$$

The LMMSE estimator is equivalent to the Bayes estimator under the Gaussian likelihood assumption with a diffuse prior [5]. Its use is justified because the state variables, position, and velocity are integrated from WNA and the FC has no prior track data, so their prior PDF is diffuse.

C. The LR Transformation of Mode Probabilities

Local mode probabilities are computed at the LT by multivariate Gaussian PDF likelihoods evaluated at the latest local measurements. Therefore, the probabilities

[2] The prior is diffuse because the state vector is composed of position and velocity only, which are integrated states of a white noise acceleration (WNA) driven model (i.e., they are nonstationary processes). Before any data arrive, the priors on these states are diffuse. If a target mode contains states that follow a stationary process, such as Ornstein–Uhlenbeck acceleration, then the acceleration is a stationary process with a proper prior. This may also be true if the **F** matrix is unique for each mode. While fusion can still be performed in a sub-optimal manner by assuming diffuse priors on the target state, the problems of accommodating switching **F** and optimally treating stationary process states will be treated in future work.

are themselves stochastic processes. Since finding a parametric joint PDF of these nonlinear transformations is not feasible, a solution is to transform the probabilities into LRs and use a multivariate Gaussian approximation of the transformed variables. This approximation is appropriate because LRs have infinite support and the multivariate Gaussian density can capture dependencies by nonzero covariances. The means variances and covariances are then readily computed as those of the difference of quadratic forms of Gaussian random variables (the innovations).

The single LR at LT j, denoted as ω_j, is selected to be the log of the ratio of the mode 1 probability to the mode 2 probability as

$$\omega_j = \ln \frac{\mu_j^1}{\mu_j^2}. \tag{19}$$

Note that only a single LR ω_j uniquely determines the probability pair, so the second log-ratio does not need to be included in the analysis of the likelihood function as it certainly does not provide additional information. If there are more than two modes, then any mode probability can serve as the common denominator for all the LRs, but the rest of this paper will concentrate on the two-mode scenario only.

The LR transformation is one-to-one, and the probabilities can be recovered using

$$\mu_j^1 = \frac{e^{\omega_j}}{e^{\omega_j}+1} \quad \mu_j^2 = \frac{1}{e^{\omega_j}+1}. \tag{20}$$

The transformation allows the new variables to be represented as a nonlinear first-order Markov process driven by the wide-sense white MCI $\boldsymbol{\nu}_j^n(k)$ (see Appendix: Property 1) with LT-computed covariances $\mathbf{S}_j^m(k)$. At the LT, the mode probabilities are computed as posterior probabilities using Gaussian likelihood functions [4]:

$$\mu_j^m(k) = \sum_{l=1}^{2} \pi^{lm} \mu_j^l(k-1) \cdot \frac{1}{c} |2\pi \mathbf{S}_j^m(k)|^{-\frac{1}{2}} e^{-\frac{1}{2}\left[\nu_j^m(k)\right]' \left[\mathbf{S}_j^m(k)\right]^{-1} \nu_j^m(k)}. \tag{21}$$

Using equation (21), the normalizing constant c is canceled in the ratio and equation (19) becomes

$$\omega_j(k) = \ln \frac{\pi^{11} e^{\omega_j(k-1)} + \pi^{21}}{\pi^{12} e^{\omega_j(k-1)} + \pi^{22}} + \frac{1}{2} \ln \frac{|\mathbf{S}_j^2(k)|}{|\mathbf{S}_j^1(k)|} + \frac{1}{2} \boldsymbol{\nu}_j^2(k)' \mathbf{S}_j^2(k)^{-1} \boldsymbol{\nu}_j^2(k) - \frac{1}{2} \boldsymbol{\nu}_j^1(k)' \mathbf{S}_j^1(k)^{-1} \boldsymbol{\nu}_j^1(k). \tag{22}$$

The means of the LR processes $\omega_1(k)$, $\omega_2(k)$ are nonzero, and they have a finite variance and nonzero correlation. The first term in equation (22), conditioned on mode n, has the first-order Taylor series expansion around $\breve{\omega}_j^{|n}(k-1)$ (the mixed initial condition—see Section IV-C)

$$\begin{aligned} &\ln \frac{\pi^{11} e^{\omega_j^{|n}(k-1)} + \pi^{21}}{\pi^{12} e^{\omega_j^{|n}(k-1)} + \pi^{22}} \\ &\approx \left[\frac{\pi^{11} e^{\breve{\omega}_j^{|n}(k-1)}}{\pi^{11} e^{\breve{\omega}_j^{|n}(k-1)} + \pi^{21}} - \frac{\pi^{12} e^{\breve{\omega}_j^{|n}(k-1)}}{\pi^{12} e^{\breve{\omega}_j^{|n}(k-1)} + \pi^{22}} \right] \omega_j^{|n}(k-1) \\ &= \left[\breve{\mu}_j^{11|n}(k-1) - \breve{\mu}_j^{12|n}(k-1) \right] \omega_j^{|n}(k-1). \end{aligned} \tag{23}$$

The $\mu_j^{lm|n}$ are the (actual) initial condition mixing weights at the local IMMs, which are functions of the LR, all conditioned on mode n:

$$\begin{aligned} \mu_j^{1m|n}(k-1) &= \frac{\pi^{1m} e^{\omega_j^{|n}(k-1)}}{\pi^{1m} e^{\omega_j^{|n}(k-1)} + \pi^{2m}}, \\ \mu_j^{2m|n}(k-1) &= \frac{\pi^{2m}}{\pi^{1m} e^{\omega_j^{|n}(k-1)} + \pi^{2m}}, \end{aligned} \tag{24}$$

and $\breve{\mu}_j^{lm|n}(k-1)$ are computed according to equation (24) by using $\breve{\omega}_j^{|n}(k-1)$ instead of $\omega_j^{|n}(k-1)$.

The last two terms of equation (22) are the difference of quadratic forms of the innovations, which are correlated between the modes and sensors. Since they are unknown to the FC and they are stochastic, they are considered to be a common additive noise for the LR of both sensors, and the mean and covariance of this noise are readily computed to form a Gaussian approximation.

Omitting k again, the mean and covariance of the LR additive noise can be derived by first defining the stacked vector of the zero mean (see Appendix: Property 1) innovations as

$$\boldsymbol{\nu}^{|n} = \left[(\boldsymbol{\nu}_1^{1|n})' \; (\boldsymbol{\nu}_1^{2|n})' \; (\boldsymbol{\nu}_2^{1|n})' \; (\boldsymbol{\nu}_2^{2|n})' \right]'. \tag{25}$$

The covariance of equation (25) is $\mathbf{S}^{|n}$, derived in Section IV-B. Together with the selection matrices

$$\begin{aligned} \mathbf{L}_1^1 &= [\mathbf{I} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0}] \quad \mathbf{L}_1^2 = [\mathbf{0} \quad \mathbf{I} \quad \mathbf{0} \quad \mathbf{0}] \\ \mathbf{L}_2^1 &= [\mathbf{0} \quad \mathbf{0} \quad \mathbf{I} \quad \mathbf{0}] \quad \mathbf{L}_2^2 = [\mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{I}] \end{aligned} \tag{26}$$

the quadratic forms can be written as

$$\begin{aligned} &(\boldsymbol{\nu}_j^{2|n})' (\mathbf{S}_j^2)^{-1} \boldsymbol{\nu}_j^{2|n} - (\boldsymbol{\nu}_j^{1|n})' (\mathbf{S}_j^1)^{-1} \boldsymbol{\nu}_j^{1|n} \\ &= (\boldsymbol{\nu}^{|n})' \left[\left(\mathbf{L}_j^2\right)' (\mathbf{S}_j^2)^{-1} \mathbf{L}_j^2 - \left(\mathbf{L}_j^1\right)' (\mathbf{S}_j^1)^{-1} \mathbf{L}_j^1 \right] \boldsymbol{\nu}^{|n} \\ &= (\boldsymbol{\nu}^{|n})' \mathbf{M}_j \boldsymbol{\nu}^{|n} \triangleq \mathbf{d}^{|n} \end{aligned} \tag{27}$$

with

$$\mathbf{M}_j \triangleq \left(\mathbf{L}_j^2\right)' (\mathbf{S}_j^2)^{-1} \mathbf{L}_j^2 - \left(\mathbf{L}_j^1\right)' (\mathbf{S}_j^1)^{-1} \mathbf{L}_j^1. \tag{28}$$

The hidden matrix $\mathbf{S}_j^m$ is computed at the LT and its expected value can be computed at the FC using the algorithm in Section IV-D and is different from the elements of $\mathbf{S}^{|n}$. The two-dimensional (2D), white,

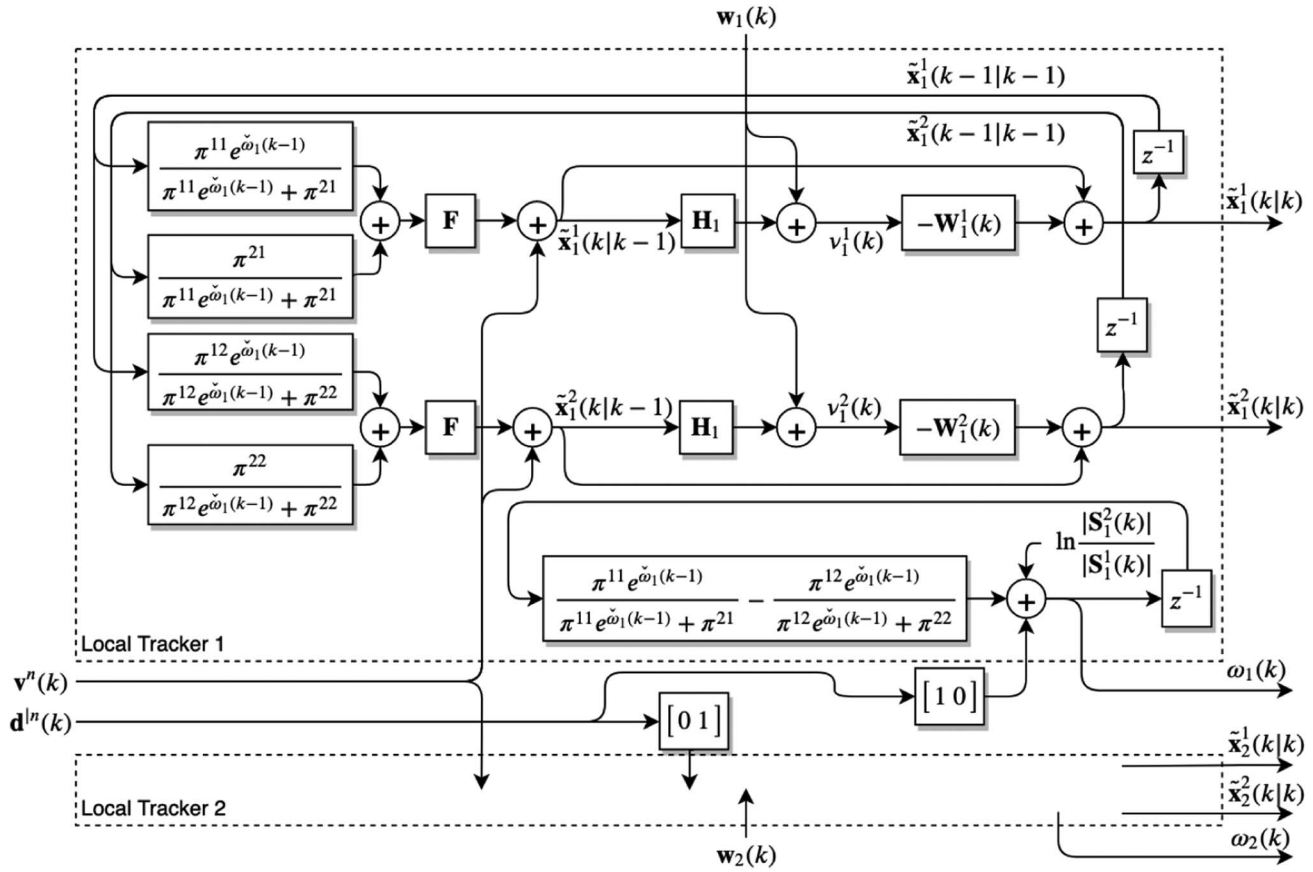


Fig. 1. Block diagram of the linearized model of the mode-conditioned errors and LRs of two IMM LTs (with only one shown explicitly) tracking the same target, from which the computation of the mode-conditioned error covariance, LR mean, and LR covariance can be derived. The evolution of the stochastic matrices $\mathbf{S}_j^m(k)$ and $\mathbf{W}_j^m(k)$ are not shown. Note that this block diagram describes the behavior of the LT but does not describe the fusion algorithm itself. z^{-1} represents a unit delay to indicate variables from the previous time step.

nonzero mean Gaussian random process $\mathbf{d}$ will approximate the quadratic form noise (27), having mean and variance/covariance found by (see [4])

$$\bar{\mathbf{d}}^{|n} \triangleq E[\mathbf{d} \mid n] = \begin{bmatrix} E\left[\boldsymbol{v}'\mathbf{M}_1\boldsymbol{v}\right] \\ E\left[\boldsymbol{v}'\mathbf{M}_2\boldsymbol{v}\right] \end{bmatrix} = \begin{bmatrix} \mathrm{tr}\left[\mathbf{M}_1\mathbf{S}^{|n}\right] \\ \mathrm{tr}\left[\mathbf{M}_2\mathbf{S}^{|n}\right] \end{bmatrix}, \quad (29)$$

$$\mathbf{D}^{|n} \triangleq E\left[(\mathbf{d}-\bar{\mathbf{d}})(\mathbf{d}-\bar{\mathbf{d}})' \mid n\right] = \begin{bmatrix} 2\mathrm{tr}\left[\left(\mathbf{M}_1\mathbf{S}^{|n}\right)^2\right] & 2\mathrm{tr}\left(\mathbf{M}_1\mathbf{S}^{|n}\mathbf{M}_2\mathbf{S}^{|n}\right) \\ 2\mathrm{tr}\left(\mathbf{M}_1\mathbf{S}^{|n}\mathbf{M}_2\mathbf{S}^{|n}\right) & 2\mathrm{tr}\left[\left(\mathbf{M}_2\mathbf{S}^{|n}\right)^2\right] \end{bmatrix}. \quad (30)$$

The covariance between a zero mean Gaussian vector and a quadratic form in the same vector is zero [15]—this means that $\bar{\mathbf{d}}$ is not correlated to the innovations, the process noise, or the measurement noise. See the Appendix for further details regarding this.

D. Fused Mode Probabilities

The received MCEs $\hat{\mathbf{x}}_j^m$ do not contain information about the target mode $M(k)$ (see Appendix: Property 4). Using Bayes' theorem and omitting the time step k for brevity, the posterior fused-mode probabilities are[3]

$$\begin{aligned} \mu_{\mathrm{F}}^n &= P\left(M = n \mid \mu_1^1, \mu_1^2, \mu_2^1, \mu_2^2\right) \\ &= P\left(M = n \mid \omega_1, \omega_2\right) \\ &= \frac{p(\omega_2 \mid \omega_1, M = n)P(M = n \mid \omega_1)}{b} \\ &= \frac{p(\omega_1, \omega_2, \mid M = n)\mu_1^n}{bp(\omega_1 \mid M = n)} \end{aligned} \quad (31)$$

with b the normalizing constant and the likelihood function of the mode based on the LR represented as

$$p(\omega_1, \omega_2 \mid M = n). \quad (32)$$

The goal here is to find the prior mean $[\bar{\omega}_1^{|n}(k)\ \ \bar{\omega}_2^{|n}(k)]'$ and the covariance $\boldsymbol{\Omega}^{|n}(k)$, conditioned under target mode n, of the Gaussian approximation of equation (32) before any data arrive. From this, the marginal in the de-

[3]The representation of equation (31) is not unique—either mode probability can be used as the prior, or the prior can be noninformative. The attractiveness of using a received probability as a prior is the ability to use as much information in the data as possible before the Gaussian approximation. In other words, the ability to directly factor in a probability as a prior can be advantageous from an accuracy perspective.

nominator of equation (31) is easily found and the likelihood can be evaluated for each mode $n = 1, 2$.

E. The LT MCEEs

Recursive covariance computations can be used to find the covariance of the zero mean MCEE. At every step k, there are *two mode hypotheses*, represented by $n = 1, 2$. As in Fig. 1, the error of the mode m prediction at tracker j, conditioned on mode n being the true mode, is

$$\begin{aligned}\tilde{\mathbf{x}}_j^{m|n}(k+1|k) &\triangleq \mathbf{F}\mathbf{x}(k) + \mathbf{v}^n(k) - \mathbf{F}\check{\mathbf{x}}_j^{m|n}(k|k) \\ &= \mathbf{F}\left[\mathbf{x}(k) - \left[\mu_j^{1m|n}\hat{\mathbf{x}}_j^{1|n}(k|k) + \mu_j^{2m|n}\hat{\mathbf{x}}_j^{2|n}(k|k)\right]\right] \\ &\quad + \mathbf{v}^n(k) \\ &= \mathbf{F}\left[\mu_j^{1m|n}\tilde{\mathbf{x}}_j^{1|n}(k|k) + \mu_j^{2m|n}\tilde{\mathbf{x}}_j^{2|n}(k|k)\right] \\ &\quad + \mathbf{v}^n(k).\end{aligned} \tag{33}$$

The MCEEs are propagated from the previous mode-conditioned prediction errors (MCPEs) as

$$\begin{aligned}\tilde{\mathbf{x}}_j^{m|n}(k|k) &= \mathbf{x}(k) - \hat{\mathbf{x}}_j^{m|n}(k|k) \\ &= (\mathbf{I} - \mathbf{W}_j^m(k)\mathbf{H}_j)\tilde{\mathbf{x}}_j^{m|n}(k|k-1) \\ &\quad - \mathbf{W}_j^m(k)\mathbf{w}_j(k).\end{aligned} \tag{34}$$

Since the LRs are system states, the weighting of the MCEE by the mixing probabilities in equation (33) is a nonlinear function of the state variables, which can be linearized by using Jacobians (see Section IV-A). Notice in equations (33) and (34) that as long as $\mathbf{F}$ is properly matched to the target's dynamics and previous errors and noise terms are zero mean, all MCEEs are zero mean.

$$E\left[\tilde{\mathbf{x}}_j^{m|n}(k|k)\right] = \mathbf{0}. \tag{35}$$

It is evident why a target whose dynamic mode switching includes changing $\mathbf{F}$ is more difficult to analyze: because $\mathbf{F}$ is common to both terms in equation (33), it can be factored out, yielding an expression in the MCEE. This is required for recursively computing the likelihood function parameters. Switching $\mathbf{F}$ requires additional analysis and algorithmic complexity to describe the evolution of nonzero mean MCEE and is beyond the scope of this paper. The Kalman gains $\mathbf{W}_j^l(k)$ are unknown to the FC directly, but expected values can be used in their place (see Section IV-D).

Note that the MCEE are never computed at the LT (the MCE are); equations (33) and (34) are only a probabilistic analysis of the errors for the purpose of finding the likelihood function parameters. Upon linearization, the MCEEs are jointly Gaussian stationary processes (see Section IV-A).

F. The System State of the IMM Trackers

The vector of the MCEEs and LRs describes the internal behavior of two IMM trackers estimating the state of the same target for the purpose of computing the required parameters of equations (12) and (32). Conditioned on mode n, it is defined as

$$\mathbf{y}^{|n}(k) \triangleq \left[\tilde{\mathbf{x}}_1^{|n}(k|k)'\ \tilde{\mathbf{x}}_2^{|n}(k|k)'\ \omega_1^{|n}(k)\ \omega_2^{|n}(k)\right]' \tag{36}$$

with the stacked vector of errors from each sensor written for compactness as

$$\tilde{\mathbf{x}}_1^{|n}(k|k) \triangleq \left[\tilde{\mathbf{x}}_1^{1|n}(k|k)'\ \tilde{\mathbf{x}}_1^{2|n}(k|k)'\right]'. \tag{37}$$

The mean of equation (36) is (considering that the MCE have zero mean error according to Section III-E)

$$\bar{\mathbf{y}}^{|n}(k) \triangleq E\left[\mathbf{y}^{|n}(k)\right] = \left[\mathbf{0}\quad \bar{\omega}_1^{|n}(k)\quad \bar{\omega}_2^{|n}(k)\right]'. \tag{38}$$

The covariance of equation (36) is

$$\mathbf{Y}^{|n}(k) \triangleq E\left[\mathbf{y}^{|n}(k)\mathbf{y}^{|n}(k)'\right] = \begin{bmatrix}\mathbf{P}^{|n}(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{\Omega}^{|n}(k)\end{bmatrix}, \tag{39}$$

where the zeros on the off-diagonal blocks are a result of the block-diagonal Jacobian and additive noise covariance to be derived in Sections IV-A and IV-B, respectively. Recursions yield $\bar{\mathbf{y}}^{|n}(k)$ and $\mathbf{Y}^{|n}(k)$, under each hypothesis $n = 1, 2$, from which the parameters of the likelihood functions, equations (12) and (32), can be computed. This will be developed in Section IV-B. These parameters are not conditioned on any previous track information, but they do require knowledge of the measurement models, the dynamic models, and the TPM.[4]

A linearized system description of two parallel IMM LTs is depicted in Fig. 1. This diagram shows the utility of the model: the white sequences, $\mathbf{v}^n(k)$ and $\mathbf{d}^{|n}(k)$, act as common inputs to both IMM subsystems, the measurement errors $\mathbf{w}_i(k)$ act as independent inputs to each IMM, and the MCEE $\tilde{\mathbf{x}}_j^m(k|k)$ and LRs $\omega_j(k)$ act as the outputs. It is the mode-conditioned, Gaussian-approximated PDF parameters of these outputs that are of interest.

IV. ALGORITHM IMPLEMENTING BAYESIAN FUSION USING IMM INSIDE INFORMATION

While the previous section discussed important preliminary fusion theory, this section develops the algorithm for fusion with IMM Inside Information.

[4]Due to the recursive algorithm, initial conditions for $\bar{\mathbf{y}}^{|n}(k)$ and $\mathbf{Y}^{|n}(k)$ do need to be provided. Standard covariance initialization methods can be used (see [4]) and the mode-conditioned mean of the LR can be initialized to zero. This can be accomplished offline.

A. The System State Transition Jacobian

From equations (33) and (34), the Jacobians of the MCPEs with respect to the previous MCEE are

$$\mathbf{J}^{\tilde{\mathbf{x}}_j^{m|n}(k+1|k)}_{\tilde{\mathbf{x}}_i^{l|n}(k|k)}(k) = \breve{\mu}_j^{lm|n}(k)\mathbf{F}\delta_{i-j}, \tag{40}$$

where δ_{i-j} is the Kronecker Delta function (i.e., the crosssensor Jacobians in equation (40) are zero).

The Jacobians of the MCPEs with respect to the previous LR, evaluated at the mean of the errors (which are zero), are zero:

$$\mathbf{J}^{\tilde{\mathbf{x}}_j^{m|n}(k+1|k)}_{\omega_j^{|n}(k)}(k) = \frac{\pi^{1m}\pi^{2m}e^{\breve{\omega}_j^{|n}}}{\pi^{1m}e^{\breve{\omega}_j^{|n}} + \pi^{2m}}\mathbf{F}E\left[\tilde{\mathbf{x}}_j^{1|n}(k|k) - \tilde{\mathbf{x}}_j^{2|n}(k|k)\right] = \mathbf{0}. \tag{41}$$

The Jacobians of the LR with respect to their previous values can be derived from equation (23). The Jacobians of the LR with respect to the previous MCE are zero since the partial derivative of the quadratic form of innovations with respect to an innovation is scaled by that innovation, which is zero mean. This is in agreement with the claim that $\mathbf{d}_j(k)$ can be treated as white, additive noise. Omitting k, the complete Jacobian is

$$\mathbf{J}^{|n} = \begin{bmatrix} \breve{\mu}_1^{11|n}\mathbf{F} & \breve{\mu}_1^{21|n}\mathbf{F} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \breve{\mu}_1^{12|n}\mathbf{F} & \breve{\mu}_1^{22|n}\mathbf{F} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \breve{\mu}_2^{11|n}\mathbf{F} & \breve{\mu}_2^{21|n}\mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \breve{\mu}_2^{12|n}\mathbf{F} & \breve{\mu}_2^{22|n}\mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{J}^{\boldsymbol{\omega}^{|n}(k+1)}_{\boldsymbol{\omega}^{|n}(k)} \end{bmatrix} \tag{42}$$

with

$$\mathbf{J}^{\boldsymbol{\omega}^{|n}(k+1)}_{\boldsymbol{\omega}^{|n}(k)} = \begin{bmatrix} \breve{\mu}_1^{11|n} - \breve{\mu}_1^{12|n} & \mathbf{0} \\ \mathbf{0} & \breve{\mu}_2^{11|n} - \breve{\mu}_2^{12|n} \end{bmatrix} \tag{43}$$

according to equation (23).

B. Recursion for the System Mode-Conditioned Means and Covariances

Having computed $\mathbf{J}^{|n}(k)$, the linearized system description for equation (36) under mode n becomes

$$\mathbf{y}^{|n}(k+1) = \mathbf{K}(k)\mathbf{J}^{|n}(k)\mathbf{y}^{|n}(k) + \mathbf{\Gamma}(k)\mathbf{g}^{|n}(k), \tag{44}$$

with

$$\begin{aligned} \mathbf{K}(k) &\triangleq \operatorname{diag}\left(\mathbf{K}_1^1, \mathbf{K}_1^2, \mathbf{K}_2^1, \mathbf{K}_2^2\right) \\ \mathbf{K}_j^m(k) &\triangleq \mathbf{I} - \mathbf{W}_j^m(k)\mathbf{H}_j. \end{aligned} \tag{45}$$

The noise vector

$$\mathbf{g}^{|n}(k) = \left[\mathbf{v}^n(k)' \ \mathbf{w}_1(k)' \ \mathbf{w}_2(k)' \ \mathbf{d}(k)'\right]' \tag{46}$$

has mean

$$\bar{\mathbf{g}}^{|n}(k) = E\left[\mathbf{g}(k) \mid M(k) = n\right] = \left[\mathbf{0} \ \bar{\mathbf{d}}^{|n}(k)'\right]', \tag{47}$$

where $\bar{\mathbf{d}}^{|n}(k)$ is defined in equation (29). The covariance of $\mathbf{g}^{|n}(k)$ is

$$\begin{aligned} \mathbf{G}^{|n}(k) &\triangleq E\left[[\mathbf{g}(k) - \bar{\mathbf{g}}^{|n}(k)][\mathbf{g}(k) - \bar{\mathbf{g}}^{|n}(k)]' \mid M(k) = n\right] \\ &= \operatorname{diag}\left[\mathbf{Q}^n, \mathbf{R}_1, \mathbf{R}_2, \mathbf{D}^{|n}(k)\right] \end{aligned} \tag{48}$$

and

$$\mathbf{\Gamma}(k) = \begin{bmatrix} \mathbf{I} - \mathbf{W}_1^1(k)\mathbf{H}_1 & \mathbf{W}_1^1(k) & \mathbf{0} & \mathbf{0} \\ \mathbf{I} - \mathbf{W}_1^2(k)\mathbf{H}_1 & \mathbf{W}_1^2(k) & \mathbf{0} & \mathbf{0} \\ \mathbf{I} - \mathbf{W}_2^1(k)\mathbf{H}_2 & \mathbf{0} & \mathbf{W}_2^1(k) & \mathbf{0} \\ \mathbf{I} - \mathbf{W}_2^2(k)\mathbf{H}_2 & \mathbf{0} & \mathbf{W}_2^2(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{49}$$

Note that $\mathbf{K}(k)\mathbf{J}^{|n}(k)$ has eigenvalues inside the complex unit circle so the recursion should always converge. So, after $\mathbf{K}(k)$, $\mathbf{J}^{|n}(k)$, and $\mathbf{\Gamma}(k)$ are computed, the mean of the LR is updated as

$$\bar{\mathbf{y}}^{|n}(k+1) = \mathbf{K}(k)\mathbf{J}^{|n}(k)\breve{\mathbf{y}}^{|n}(k) + \mathbf{\Gamma}(k)\bar{\mathbf{g}}^{|n}(k), \tag{50}$$

where $\breve{\mathbf{y}}^{|n}(k)$ is a *mixed initial condition of the system recursion* with covariance $\breve{\mathbf{Y}}^{|n}(k)$.

The system's covariance update is

$$\begin{aligned} &\mathbf{Y}^{|n}(k+1) \\ &= \mathbf{K}(k)\mathbf{J}^{|n}(k)\breve{\mathbf{Y}}^{|n}(k)\mathbf{J}^{|n}(k)'\mathbf{K}(k)' + \mathbf{\Gamma}(k)\mathbf{G}^{|n}(k)\mathbf{\Gamma}(k)'. \end{aligned} \tag{51}$$

The first block on the diagonal of equation (51) is the covariance of the MCEE. With[5]

$$\mathbf{P}^{|n}(k+1|k) \triangleq \left[\mathbf{J}^{|n}(k)\breve{\mathbf{Y}}^{|n}(k)\mathbf{J}^{|n}(k)'\right]_1^{4N_x} \tag{52}$$

representing the covariance of the local MCPE, where only the first $4N_x$ rows and columns of $\mathbf{J}^{|n}(k)\breve{\mathbf{Y}}^{|n}(k)\mathbf{J}^{|n}(k)'$ are selected, the covariances and crosscovariances of the MCIs are computed as

$$\mathbf{S}^{|n}(k+1) = \mathbf{H}\mathbf{P}^{|n}(k+1|k)\mathbf{H}' + \mathbf{L}_{\mathbf{w}}\begin{bmatrix} \mathbf{R}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_2 \end{bmatrix}\mathbf{L}_{\mathbf{w}}' \tag{53}$$

with

$$\mathbf{H} = \operatorname{diag}(\mathbf{H}_1, \mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_2), \tag{54}$$

$$\mathbf{L}_{\mathbf{w}} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{I} \end{bmatrix}'. \tag{55}$$

[5]The notation $k+1|k$ used in covariances computed at the FC serves only to show that they are related to the state predictions made at the LT. It is not intended to mean that the computations at the FC are conditioned on past measurements.

C. The Mixing Process for Hypothesis Merging

Before every recursion update step (50)–(51), the hypotheses from the previous step must be merged, just as they are in the IMM estimation algorithm. In the absence of any previous observations, the time-invariant mixing probabilities are computed at the FC from the steady-state Markov chain probabilities $\mu^1_\infty, \mu^2_\infty$ as priors [4]:

$$\mu^{ln} = \frac{\pi^{ln}\mu^l_\infty}{\pi^{1n}\mu^1_\infty + \pi^{2n}\mu^2_\infty}. \tag{56}$$

With $M(k+1) = n$ the event that the next mode is n, the mixed initial conditions are

$$\check{\mathbf{y}}^{|n}(k) = E\left[\mathbf{y}(k) \mid M(k+1) = n\right] = \sum_{l=1}^{2} \mu^{ln}\bar{\mathbf{y}}^{|l}(k), \tag{57}$$

$$\begin{aligned}
&\check{\mathbf{Y}}^{|n}(k) \\
&= E\left[\left[\mathbf{y}(k) - \check{\mathbf{y}}^{|n}(k)\right]\left[\mathbf{y}(k) - \check{\mathbf{y}}^{|n}(k)\right]' \mid M(k+1) = n\right] \\
&= \sum_{l=1}^{2} \mu^{ln}\left[\mathbf{Y}^{|l}(k) + \left[\bar{\mathbf{y}}^{|l}(k) - \check{\mathbf{y}}^{|n}(k)\right]\left[\bar{\mathbf{y}}^{|l}(k) - \check{\mathbf{y}}^{|n}(k)\right]'\right].
\end{aligned} \tag{58}$$

D. Expected Values of the LT Stochastic Matrices

The local mode-conditioned innovation covariances $\mathbf{S}^m_j(k)$ and Kalman gains $\mathbf{W}^m_j(k)$ are required for equations (27), (45), and (49). To proceed, it should be first noted that both $\mathbf{S}^m_j(k)$ and $\mathbf{W}^m_j(k)$ are stochastic matrices, computed from a mixed covariance matrix that includes the spread-of-the-means (SOM) of the Gaussian mixture. The SOM results in a covariance update that is measurement-dependent [4]. Because the local MCE and mode probabilities depend on these stochastic matrices, the covariances as computed by the LT IMM algorithm behave like "state variables" of the system and are recursively updated. The expected value of each matrix can be computed through linearization, mixing, and recursion, then treating the resulting matrices as having zero variance (see Appendix: Property 8). This can be accomplished by expanding the FC recursion process to include finding the matrix means of the mixed initial condition covariances $\bar{\check{\mathbf{P}}}^m_j(k-1|k-1)$ and using them to find $\bar{\mathbf{S}}^m_j(k)$ and $\bar{\mathbf{W}}^m_j(k)$ using standard Kalman equations.

First, it is noted that mixed initial conditions $\check{\mathbf{x}}^m_j(k|k)$ and $\check{\mathbf{P}}^m_j(k|k)$ do not depend on the state or mode at $k+1$. Then, the recursion for the matrix mean of the local mixed initial condition matrix can be linearized as

$$\begin{aligned}
\bar{\check{\mathbf{P}}}^m_j(k|k) &\triangleq E\left[\check{\mathbf{P}}^m_j(k|k)\right] \\
&= E\left\{\sum_{l=1}^{2} \mu^{lm}_j(k)\left[\mathbf{P}^l_j(k|k) + \mathbf{N}(k)\right]\right\} \\
&= \sum_{n=1}^{2} E\left\{\sum_{l=1}^{2} \mu^{lm}_j(k)\Big[\mathbf{P}^l_j(k|k) \right. \\
&\qquad \left. + \mathbf{N}^{lm}_j(k) \mid M(k) = n\Big]\right\} \cdot P\left[M(k) = n\right] \\
&\approx \sum_{n=1}^{2}\Big[\sum_{l=1}^{2} \check{\mu}^{lm|n}_j(k) E\left[\mathbf{P}^l_j(k|k) \mid M(k) = n\right] \\
&\qquad + \check{\mu}^{lm|n}_j(k) E\left[\mathbf{N}^{lm}_j(k) \mid M(k) = n\right]\Big]\mu^n_\infty.
\end{aligned} \tag{59}$$

In equation (59), $\mu^{lm}_j(k)$ can be taken out of the expectation (as a first-order linear approximation) and evaluated using $\check{\omega}^{|n}_j(k)$ and equation (24); $\mathbf{N}^{lm}_j(k)$ is the SOM. The expectation in the first term can be computed by

$$\begin{aligned}
\bar{\mathbf{P}}^l_j(k|k) &\triangleq E\left[\mathbf{P}^l_j(k|k) \mid M(k) = n\right] = E\left[\mathbf{P}^l_j(k|k)\right] \\
&= \mathbf{F}\bar{\check{\mathbf{P}}}^l_j(k-1|k-1)\mathbf{F}' + \mathbf{Q}^l - \mathbf{W}^l_j(k)\mathbf{S}^l_j(k)\mathbf{W}^l_j(k)'.
\end{aligned} \tag{60}$$

Omitting the step k, the expected value of the SOM can be derived starting with

$$\begin{aligned}
&E\left[\mathbf{N}^{lm}_j \mid M(k) = n\right] \\
&\qquad = E\left[\left[\hat{\mathbf{x}}^l_j - \check{\mathbf{x}}^m_j\right]\left[\hat{\mathbf{x}}^l_j - \check{\mathbf{x}}^m_j\right]' \mid M(k) = n\right]
\end{aligned} \tag{61}$$

and expanding the difference as

$$\begin{aligned}
\hat{\mathbf{x}}^l_j - \check{\mathbf{x}}^m_j &= \hat{\mathbf{x}}^l_j - \sum_{o=1}^{2} \mu^{om}_j \hat{\mathbf{x}}^o_j \\
&= \begin{cases} \mu^{2m}_j\left[\hat{\mathbf{x}}^1_j - \hat{\mathbf{x}}^2_j\right] & \text{if } l = 1 \\ \mu^{1m}_j\left[\hat{\mathbf{x}}^2_j - \hat{\mathbf{x}}^1_j\right] & \text{if } l = 2. \end{cases}
\end{aligned} \tag{62}$$

Since the two MCEs have the same mean, $\mathbf{x}$,

$$\begin{aligned}
&E\left[\mathbf{N}^{lm}_j \mid M(k) = n\right] \\
&= \begin{cases} (\check{\mu}^{2m}_j)^2\left(\mathbf{P}^{11|n}_{jj} + \mathbf{P}^{22|n}_{jj} - \mathbf{P}^{12|n}_{jj} - \mathbf{P}^{21|n}_{jj}\right) & \text{if } l = 1 \\ (\check{\mu}^{1m}_j)^2\left(\mathbf{P}^{11|n}_{jj} + \mathbf{P}^{22|n}_{jj} - \mathbf{P}^{12|n}_{jj} - \mathbf{P}^{21|n}_{jj}\right) & \text{if } l = 2 \end{cases}
\end{aligned} \tag{63}$$

where each $\mathbf{P}^{lm|n}_{jj}$ is a respective block of equation (14). Finally, after computing $\bar{\check{\mathbf{P}}}^m_j(k-1|k-1)$ using equations

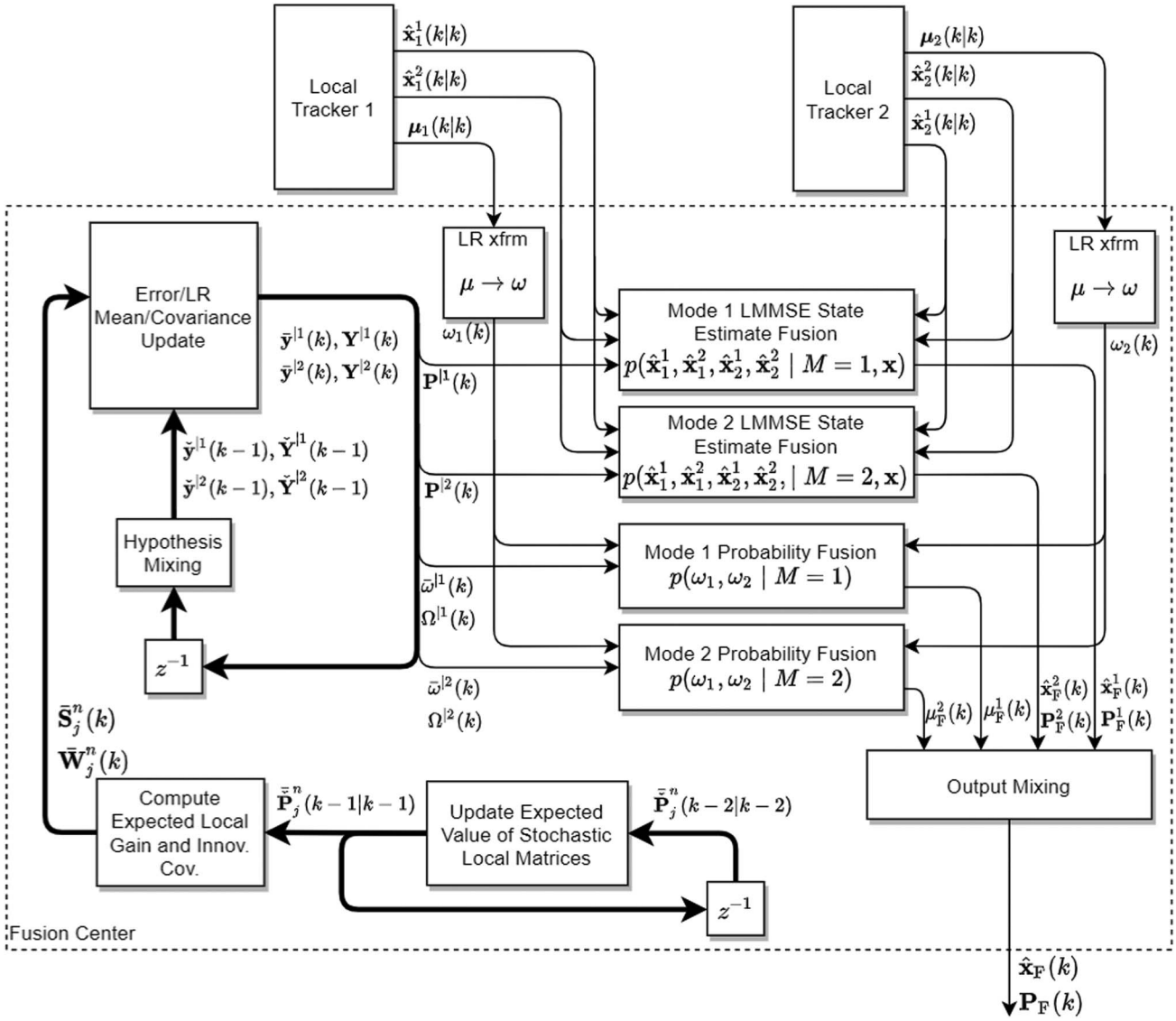


Fig. 2. Block diagram of the proposed IMM inside information fusion algorithm. $\bar{\omega}^{|n}$ is part of $\mathbf{y}^{|n}$ and $\mathbf{Y}^{|n}$ is block-diagonal with $\mathbf{P}^{|n}$ and $\mathbf{\Omega}^{|n}$ as its respective blocks. The relevant Gaussian likelihood functions used for fusion are shown in the fusion blocks.

(59), (60), and (63), the required expected values of the stochastic matrices can be computed as

$$\bar{\mathbf{P}}_j^m(k|k-1) = \mathbf{F}\bar{\bar{\mathbf{P}}}_j^m(k-1|k-1)\mathbf{F}' + \mathbf{Q}^m, \tag{64}$$

$$\bar{\mathbf{S}}_j^m(k) = \mathbf{H}_j\bar{\mathbf{P}}_j^m(k|k-1)\mathbf{H}_j' + \mathbf{R}_j, \tag{65}$$

$$\bar{\mathbf{W}}_j^m(k) = \bar{\mathbf{P}}_j^m(k|k-1)\mathbf{H}_j'\bar{\mathbf{S}}_j^m(k)^{-1}. \tag{66}$$

E. The Fused Estimate and Covariance Output

Omitting k for brevity, the fused estimate of the target state, in the minimum MSE sense, is the mean of the posterior fused mixture (9):

$$\hat{\mathbf{x}}_{\mathrm{F}} = \sum_{n=1}^{2} \mu_{\mathrm{F}}^n \hat{\mathbf{x}}_{\mathrm{F}}^n \tag{67}$$

and has expected MSE

$$\mathbf{P}_{\mathrm{F}} = \sum_{n=1}^{2} \mu_{\mathrm{F}}^n \left[\mathbf{P}_{\mathrm{F}}^n + (\hat{\mathbf{x}}_{\mathrm{F}}^n - \hat{\mathbf{x}}_{\mathrm{F}})(\hat{\mathbf{x}}_{\mathrm{F}}^n - \hat{\mathbf{x}}_{\mathrm{F}})'\right]. \tag{68}$$

V. SUMMARY

A. Algorithmic Steps

A block diagram of the overall fusion method is depicted in Fig. 2. The outside information fusion, which fuses the moment-matched IMM outputs, is depicted in Fig. 3 for comparison.

The algorithm can be interpreted as an IMM combined with an extended Kalman filter (EKF) whose recursion is executed without track data. The goal of the recursion is to compute the mode-conditioned covariance/crosscovariance matrix of the MCE (conditioned on $\mathbf{x}$) and the mode-conditioned mean plus

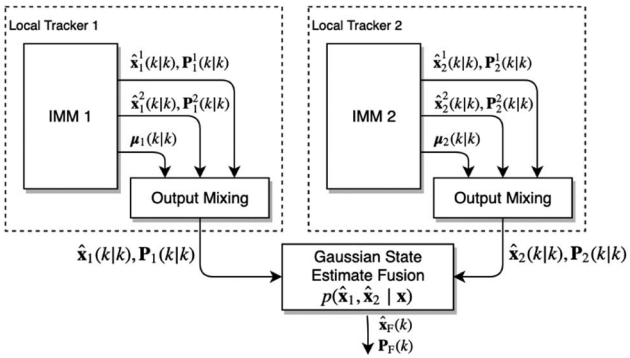


Fig. 3. Block diagram of conventional outside fusion method. The Gaussian likelihood function used is written in the fusion block.

covariance/crosscovariance matrix of the LR to be used in the fusion process when the data arrive. The state variables of the EKF/IMM-like algorithm are considered to be the MCEE $\tilde{\mathbf{x}}_j^m$, LR ω_j, and the the local covariance matrices $\mathbf{P}_j^m$ (where only the mode-conditioned mean of the components of the latter matrices is computed). The steps are as follows.

For $n = 1, 2$, initialize the following:

1) Set $\bar{\omega}_1^{|n}(0) \leftarrow 0, \bar{\omega}_2^{|n}(0) \leftarrow 0$.
2) $\mathbf{\Omega}^{|n}(0) \leftarrow c\mathbf{I}$ (where c should be large enough to cover the expected error — see [4]; a value of 1 is used in the simulations of Section VI).
3) Diagonal blocks of $\mathbf{P}^{|n}(0)$ (the covariances of the MCEE) are initialized the same as LT (see [4]), or using $c\mathbf{I}$, where c is large enough to cover the expected errors.
4) The off-diagonal blocks of $\mathbf{P}^{|n}(0)$ are initialized with zeros (i.e., the crosscovariances can be initialized to zero).
5) The expected values of the hidden local mixed initial condition matrices $\bar{\mathbf{P}}_j^n(0|0)$, for $j = 1, 2$, are initialized the same as LT and the diagonal blocks of $\mathbf{P}^{|n}(0)$.
6) Compute steady-state Markov chain probabilities μ_∞^n.
7) Set $k = 1$.

Repeat the following for each k (synchronously with LT or offline) for $j = 1, 2, l = 1, 2, m = 1, 2$, and $n = 1, 2$:

1) Mixing: Compute $\check{\mathbf{y}}^{|n}(k-1)$ and $\check{\mathbf{Y}}^{|n}(k-1)$ using equations (57) and (58).
2) Compute $\check{\mu}_j^{lm|n}(k-1)$ using equation (24) (substitute $\check{\omega}_j^{|n}(k-1)$ for $\omega_j^{|n}(k-1)$).
3) Compute the mean of the local mixed initial condition matrices $\bar{\tilde{\mathbf{P}}}_j^m(k-1|k-1)$ using equation (59).
4) Compute the expected value of the local $\bar{\mathbf{S}}_j^m(k)$ and $\bar{\mathbf{W}}_j^m(k)$ using equations (64), (65) and (66).
5) Compute joint system state mean $\bar{\mathbf{y}}^{|n}(k)$ using equation (50).
6) Compute joint system state covar $\mathbf{Y}^{|n}(k)$ using equation (51).

If LT data (10) arrive at time step k, then for $n = 1, 2$:

1) extract $\bar{\omega}_1^{|n}, j = 1, 2$, from $\bar{\mathbf{y}}^{|n}(k)$ and $\mathbf{P}^{|n}(k)$, $\mathbf{\Omega}^{|n}(k)$ from $\mathbf{Y}^{|n}(k)$. See equation (38) and (39);
2) compute fused MCE $\hat{\mathbf{x}}_F^n(k)$ using equation (15);
3) compute fused MCC $\mathbf{P}_F^n(k)$ using equation (16);
4) compute fused mode probabilities $\mu_F^n(k)$ using equation (31);
5) compute output mean $\hat{\mathbf{x}}_F(k)$ and MSE $\mathbf{P}_F(k)$ using equations (67) and (68).

B. Computational Complexity

The algorithm described in this paper consists of two main routines, which are both computationally feasible for real-time performance. The first is the recursive computation of $\bar{\mathbf{y}}^{|n}(k)$, $\mathbf{Y}^{|n}(k)$ and the hidden stochastic matrices $\bar{\mathbf{S}}_j^m(k)$ and $\bar{\mathbf{W}}_j^m(k)$. This algorithm is analogous to an IMM track predictor with EKF mode predictors (i.e., an IMM with no data). Since this paper does not explicitly generalize the algorithm to more than two modes and two LTs, the order of complexity will be discussed in terms of the state dimension N_x only. It can be seen in equations (57) and (58) that there are $2^2 = 4$ mixing operations as in a two-mode IMM, where each mixing operation scales quadratically with N_x due to the outer product of equation (58). Each mode is approximately on the order of $O[(2 \cdot 2N_x)^2]$ due to the FC's EKF-like prediction recursion (50), (51). The hidden matrix computations (59)–(66), where a matrix inversion is involved, scale approximately as $O[2 \cdot 2N_x^3]$, so the overall computational order is $O[N_x^3]$. Since the target and LT parameters are time-invariant, these recursive computations can be performed offline.

The second routine is the actual fusion of the data when they arrive. Since Gaussian LMMSE fusion is performed for each mode (see equations (15) and (16)) and matrix inversion dominates the order, its complexity scales approximately as $O[N_x^3]$.

Chernoff fusion scales similarly for each selection of the fusion exponent w due to the matrix inversion of the least-squares parameter estimation [8]. The search for the optimal w using a grid of N_w values on the interval (0,1) means that the sigma point Chernoff fusion method of [8] scales approximately as $O[N_w N_x^3]$.

VI. SIMULATION RESULTS

The simulations have the local IMM estimators tracking a target in 2D space observed by sensors that are measuring its Cartesian position. Two LTs run IMM estimators and use two dynamic modes described in the sequel. Three scenarios are considered: the first has a deterministic target trajectory (ground truth) using a

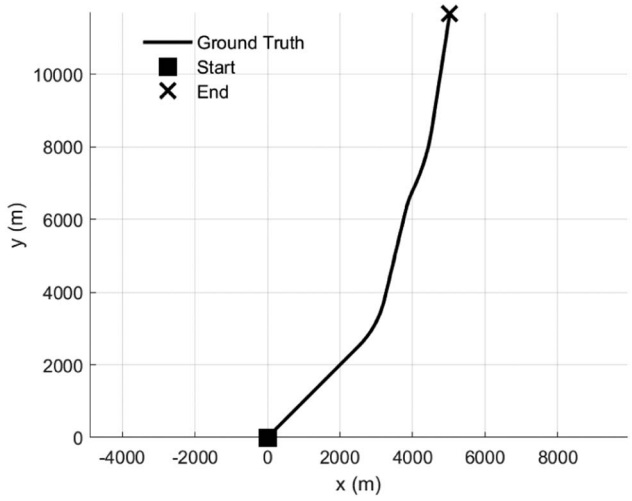


Fig. 4. The fixed target trajectory of the first scenario.

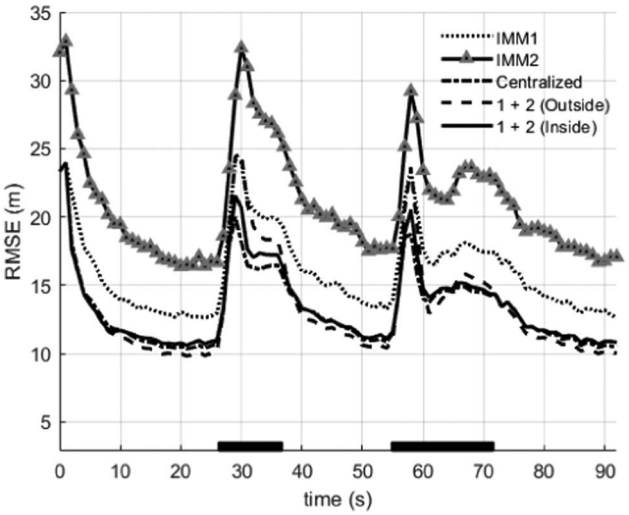


Fig. 5. Position RMSE of local IMMs, CMF, outside fusion, and inside fusion.

coordinated-turn model (see [4]) and fusion at full rate; the second has the same deterministic trajectory with fusion at a reduced rate (as a sanity check since the proposed method's performance at fusion times is not affected by fusion rate); and the third simulates random trajectories driven by white noise, with a dynamic model matching that of the estimators and with fusion at full rate.

A. Ground Truth

The measurement interval is $T = 1\,\text{s}$. The target starts at $x = 0, y = 0$ with $\dot{x} = 100\,\text{m/s}, \dot{y} = 100\,\text{m/s}$. The target:

1) travels straight for 25.2 s;
2) performs a constant-rate left turn of $3\,^\circ/\text{s}$ for 10.6 s;
3) travels straight for 18 s;
4) performs a constant-rate right turn of $-3\,^\circ/\text{s}$ for 4.1 s;
5) performs a constant-rate left turn of $1.3\,^\circ/\text{s}$ 12.8 s;
6) travels straight for 22.6 s.

A plot of this constant-speed, variable-turn rate trajectory is shown in Fig. 4.

B. Estimation Models

The state vector is composed of stacked position and velocity

$$\mathbf{x}(k) = [x(k)\; y(k)\; \dot{x}(k)\; \dot{y}(k)]'. \tag{69}$$

The estimator dynamic models are described as follows. Mode 1 is a 2D WNA model, discretized from the continuous-time model [4]. It has a 2D process noise acceleration with intensity (power spectral density) $\tilde{q}^1 = 0.01^2\,\text{m}^2/\text{s}^3$, and Mode 2 is the same but with $\tilde{q}^2 = 7.5^2\,\text{m}^2/\text{s}^3$:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{70}$$

$$\mathbf{Q}^n = \begin{bmatrix} \frac{1}{3}T^3 & 0 & \frac{1}{2}T^2 & 0 \\ 0 & \frac{1}{3}T^3 & 0 & \frac{1}{2}T^2 \\ \frac{1}{2}T^2 & 0 & T & 0 \\ 0 & \frac{1}{2}T^2 & 0 & T \end{bmatrix} \tilde{q}^n. \tag{71}$$

The TPM is

$$\Pi = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix}. \tag{72}$$

The measurement parameters are

$$\mathbf{H}_1 = \mathbf{H}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \tag{73}$$

$$\mathbf{R}_1 = \text{diag}[(15\,\text{m})^2, (18\,\text{m})^2], \tag{74}$$

$$\mathbf{R}_2 = \text{diag}[(20\,\text{m})^2, (25\,\text{m})^2]. \tag{75}$$

C. Fusion Results

Figs. 5 and 6 show the position and velocity RMSE for the inside information fusion, outside information fusion (naive Gaussian fusion with no crosscovariances), and CMF methods, along with the RMSE of the local sensor tracks. The inside fusion is slightly outperformed by the outside fusion during straight-line motion, just as the centralized fusion is, but inside fusion significantly outperforms outside fusion during maneuvers. It is interesting to note the low RMSE during

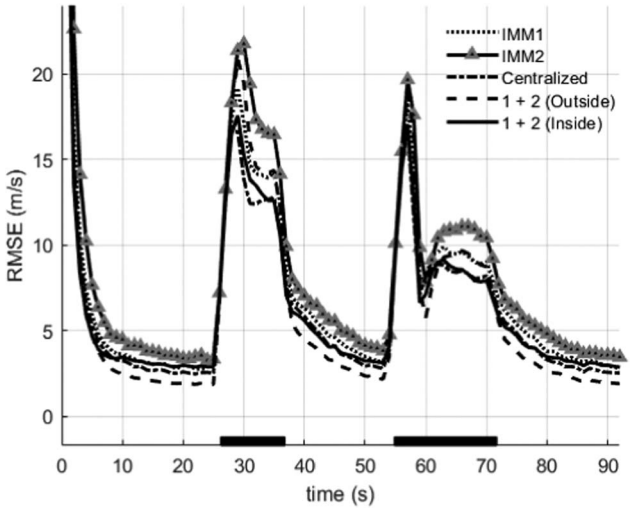


Fig. 6. Velocity RMSE of local IMMs, CMF, outside fusion, and inside fusion.

straight-line motion for naive outside fusion and that it outperforms the method of this paper and CMF during these times. This is explained by the fact that outside fusion computes a small covariance matrix that significantly reduces the overall filter "bandwidth" [4], but because of this, it performs very poorly during maneuvers (i.e., it does not minimize the maximum error). Section VI-E performs simulations with Monte Carlo randomly generated trajectories that match the stochastic model of the target, and those simulations show that, on average, CMF and the fusion with IMM inside information do indeed outperform naive outside fusion.

The consistency is evaluated using the normalized estimation error squared (NEES, see [4]), divided by N_x (the state dimension) and this is plotted for every time point in Fig. 7. Values near 1 are ideal and reflect a chi-square quadratic form resulting from estimation errors

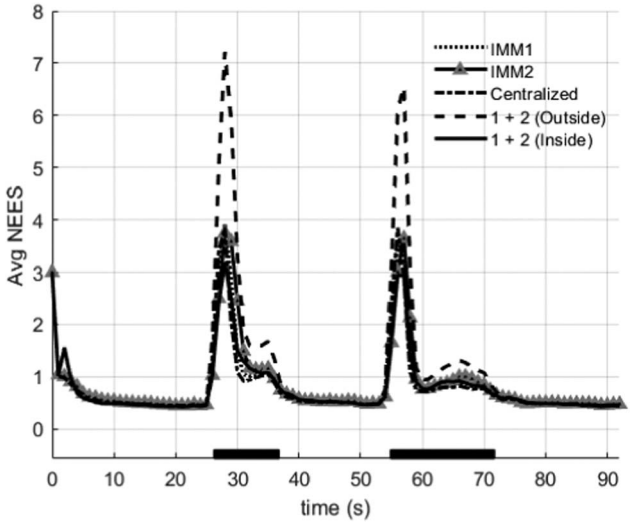


Fig. 7. NEES of local IMMs, CMF, outside fusion, and inside-information fusion. Value is normalized by N_x to be 1 when fused covariance matches sample MSE.

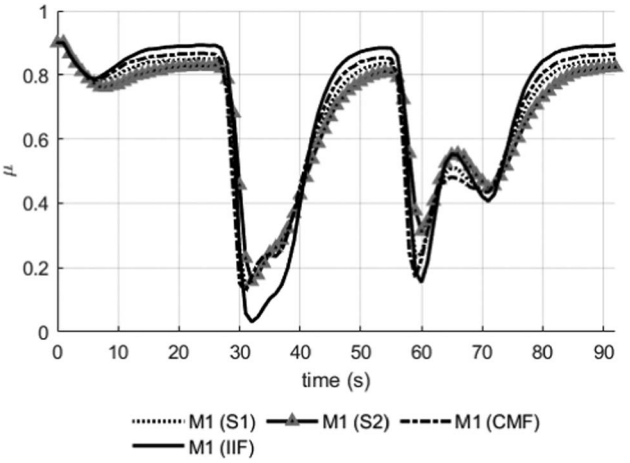


Fig. 8. Computed probability of mode 1 from local IMMs (S1, S2), inside information-fusion, and CMF.

that are zero mean and consistent with the state covariances. It is clear that the inside-information method achieves better consistency.

In Fig. 8, the mode probabilities of the inside-information fusion are compared to the local IMM mode probabilities and the CMF mode probabilities. The fused mode probabilities computed by the inside-information fusion slightly lead the probabilities of the local sensors when transitioning modes and, so, maneuvers can be detected quicker than they can be at the LT. Although the transient performance is encouraging, it can also be seen that the method as described in this paper results in fused mode probabilities that are "more sure" about the mode—centralized fusion is more conservative and only boosts this conviction slightly.

D. Reduced-Rate Fusion

As a sanity check, it should be shown that fusion performance is not affected by the rate at which track data are transmitted. An advantage to the T2TF using inside information presented here is that it is not affected by previous tracks. Outside information fusion is known not to be affected by fusion rate because it utilizes the standard Gaussian fusion method without memory. As can be seen in Fig. 2, the LR mean/covariance and the MCEE covariance (including crosscovariances between trackers) are recursively updated whether there is track information or not, and received tracks are not used in that computation (in the scenario presented here, $\bar{\mathbf{y}}^{|n}$ and $\mathbf{Y}^{|n}$ can even be computed offline).

Figs. 9–11 show the comparison of outside information fusion to the inside information fusion at a reduced rate of once every five measurement intervals, starting at $k = 4$. Looking closely, the performance at the fusion times matches the performance shown in Figs. 5–7. Again, it can be seen that the inside information fusion, like CMF, only outperforms outside information fusion during maneuvers, but the consistency of the fused covariance is significantly superior for inside-information

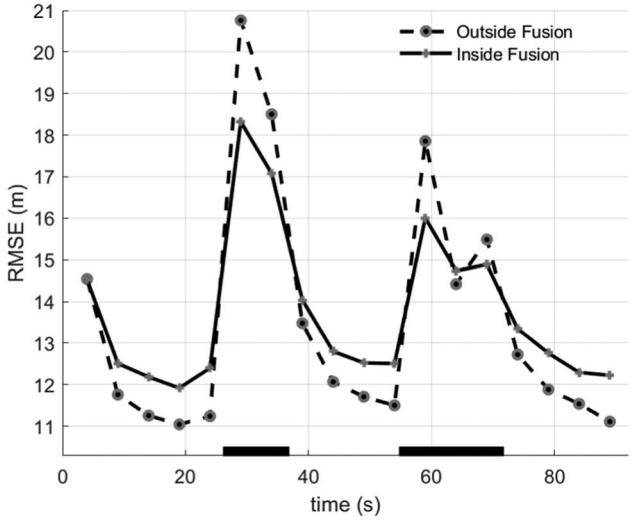


Fig. 9. Position RMSE of outside information fusion and inside information fusion at reduced rate, fusing tracks once every five measurement intervals.

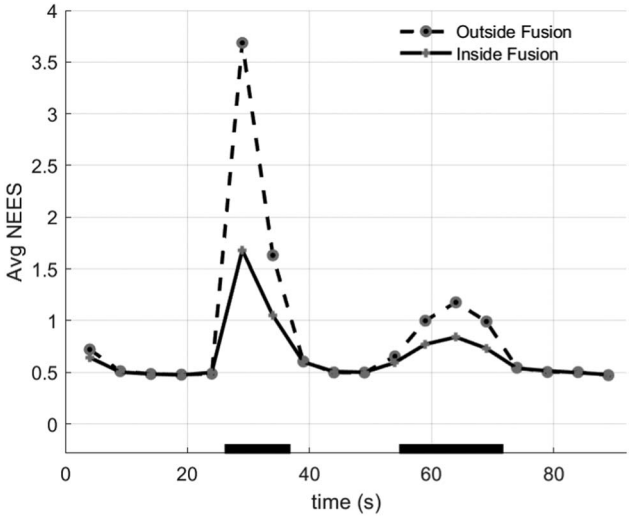


Fig. 11. NEES of outside information fusion and inside information fusion at reduced rate, fusing tracks once every five measurement intervals.

fusion. The next section demonstrates that, on average, CMF and fusion with inside information are actually more accurate.

E. Simulations Using Random, Model-Matched Target Trajectories

While the simulations of the previous sections were carried out using a single realization of the true target trajectory, the trajectory simulations of this section are randomized for every Monte Carlo run. This provides a better comparison of the overall behavior of the algorithms and highlights the consistency of the inside-information fusion.

The random trajectories are created using the WNA model driven by zero mean white noise having covariance given in equation (71).[6] The mode n is selected according to realizations of the Markov chain having TPM (72). The results are shown in Figs. 12–14. It can be seen that the position RMSE is close to equal for CMF, outside fusion, and inside fusion. Fusion with outside information has more velocity RMSE. Due to the matched target and estimator parameters, the NEES, normalized to nominal one, measures the overall MSE consistency of the IMM trackers (where CMF is simply an IMM with stacked measurement vectors). Fusion with inside information can be seen to be as consistent as centralized fusion, demonstrating that it is a fusion that accounts for error correlations (i.e., the crosscovariances) and provides a consistent fused covariance output. Outside-information fusion has NEES that is 50% higher than the ideal NEES of inside-information fusion meaning that the fused estimate covariance from outside fusion is, on average, 33% smaller than it should be given the actual sample error covariance.

Figs. 12–14 also show the results of Chernoff fusion using the sigma-point method of [8], where the weight parameter is searched at every time step in the interval [0.01, 0.99] using increments of 0.01. It can be seen that although the Chernoff fused covariance closely matches the actual sample MSE of the fused estimate (according to the NEES), its RMSE is higher than both inside-information and outside-information fusion. So in this application, the MSE consistency of Chernoff fusion is evident, but its inability to incorporate the known system parameters renders it more inaccurate than the model-based fusion using IMM inside information. On average, fusion with IMM inside information performed

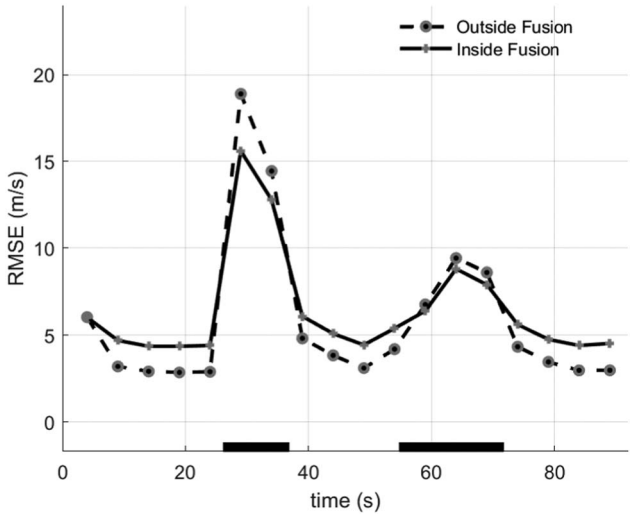


Fig. 10. Velocity RMSE of outside information fusion and inside information fusion at reduced rate, fusing tracks once every five measurement intervals.

[6]White noise is a requirement for the state of the system to be a Markov process, which is a requirement for the existence of an estimator [4].

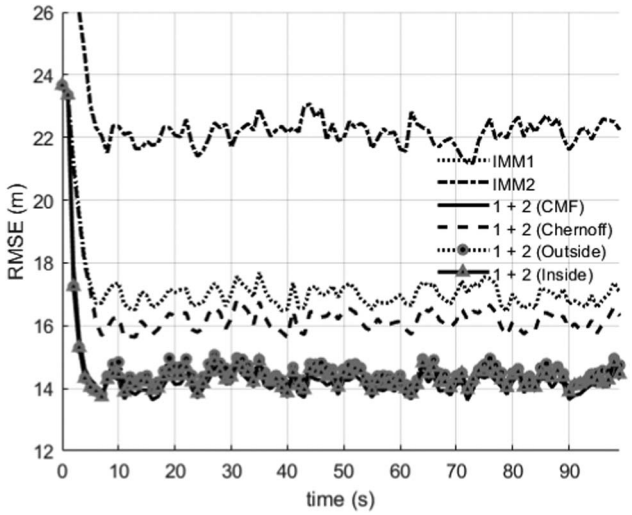


Fig. 12. Position RMSE of local IMMs, CMF, Chernoff fusion, outside fusion, and inside fusion, using random, model-matched target trajectories.

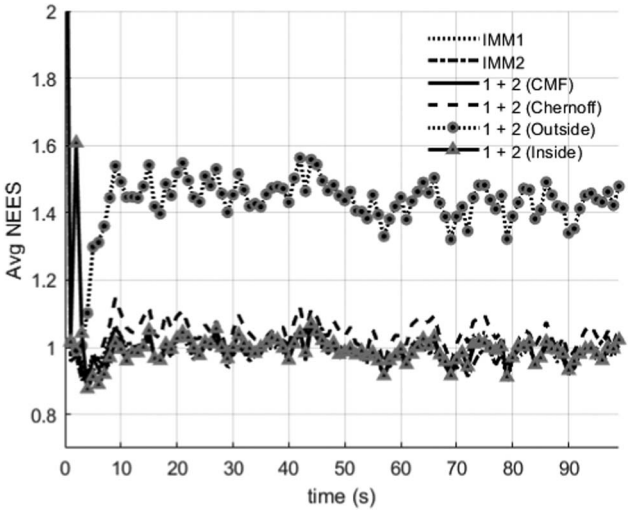


Fig. 14. NEES of local IMMs, CMF, Chernoff fusion, outside fusion, and inside fusion, using random, model-matched target trajectories. All estimators, except outside fusion, achieve ideal consistency of the MSE.

approximately 100 times faster than Chernoff fusion (see Section V-B for complexity analysis).

VII. CONCLUSIONS

A system model of two IMM trackers estimating the state of a maneuvering target was presented for T2TF using information from inside the local IMM estimators. The fusion estimator produces a posterior fused mean and covariance that is reduced from a Gaussian mixture. The mixture parameters are computed from IMM track information from two LTs, with the target modeled as jumping between two dynamic modes. The linearized system model, together with the LR transformation of the received mode probabilities, yields covariances and crosscovariances of the local mode-conditioned errors, and also yields the mode-conditioned means, variances, and covariances of the scalar LRs. From these, the parameters of the likelihood functions of the mode-conditioned state and the mode are derived. Each fused mode-conditioned state estimate uses information from all received MCE. The result is on-demand Bayesian fusion capability with no previous fused track information needed. Compared to the naive fusion of moment-matched Gaussian track information (i.e., outside information fusion), the new method achieves performance closer to the CMF method and outperforms the naive fusion in both RMSE and covariance consistency, most notably when the target is in a maneuvering mode. Fusion with inside information was shown to be consistent on average as it accounts for the crosscovariance of the local estimate errors and mode probabilities, whereas fusion with outside information and no crosscovariance has a computed covariance that is 33% too small on average. Compared to Chernoff fusion, the method is more accurate, consistent, provides mode inference, and is computationally faster.

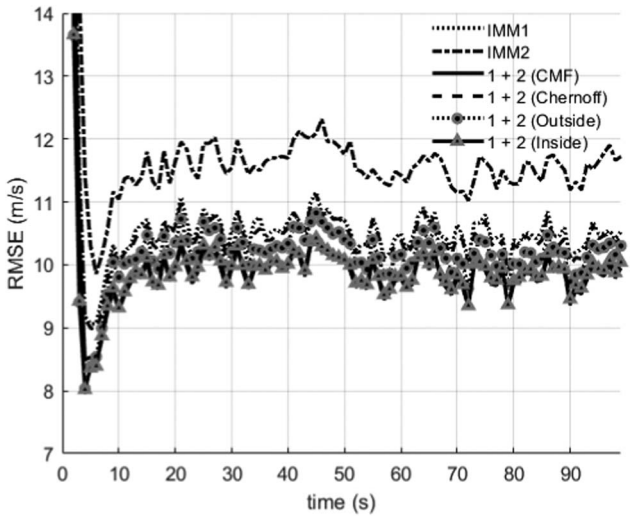


Fig. 13. Velocity RMSE of local IMMs, CMF, Chernoff fusion, outside fusion, and inside fusion, using random, model-matched target trajectories.

APPENDIX CORRELATION, DEPENDENCY, AND STATISTICAL INFORMATION PROPERTIES

The following properties of the LT track parameters and fused estimates establish some of the claims made in this paper. For jointly Gaussian densities, uncorrelated random variables are independent. Equivalently, if there is no linear dependency among them, then the variables are uncorrelated. The following claims are proved under the multiple model, linear-Gaussian approximation (Property 4 is proved without approximation):

Property 1: *The locally computed, mode-conditioned innovations are zero mean and wide-sense white sequences regardless of the true target mode*. First, it is noted that since the **F** is the same for both modes and

the process noise is zero mean, all MCEE and innovations are zero mean (see Section III-E). The whiteness proof for single-mode systems is given in [4] using the smoothing property of expectations. The following generalizes that proof because conditioning on mode $M(k)$ does not change this result. Let $k_1 < k_2$, then

$$\begin{aligned} &E\left[\boldsymbol{\nu}_j^m(k_2)\boldsymbol{\nu}_j^m(k_1)' \mid M(k_2)=n\right] \\ &\quad = E\left[E\left[\boldsymbol{\nu}_j^m(k_2)\boldsymbol{\nu}_j^m(k_1)' \mid \mathbf{Z}_j^{k_2-1}\right] \mid M(k_2)=n\right] \\ &\quad = E\left[E\left[\boldsymbol{\nu}_j^m(k_2) \mid \mathbf{Z}_j^{k_2-1}\right]\boldsymbol{\nu}_j^m(k_1)' \mid M(k_2)=n\right] \\ &\quad = \mathbf{S}_j^m(k_2)\delta_{k_2-k_1} \end{aligned} \tag{A1}$$

Property 2: *The LRs of the local mode probabilities are orthogonal to the local MCEEs*. Under the multiple-model, linear-Gaussian approximation, the LR evolve as a Markov process (22) driven by white noise that is uncorrelated to the noise that drives the MCEE process. In Section III-C, the LR Gaussian white noise $\mathbf{d}(k)$ is approximated from the quadratic form of the innovations. Likewise, conditioned on the target dynamic mode, the MCEE evolve as a Markov process according to equations (33) and (34), where their additive white noise is a linear combination of the process noise and the measurement noise. Since the covariance between a vector and its quadratic form is zero, the additive noises of the LR and MCEE are uncorrelated. Additionally, the Jacobian (42) is block-diagonal, where the MCEE do not linearly depend on previous values of the LR and vice-versa.

Property 3: *Given the LT MCEs, the LRs of the received mode probabilities do not contribute additional information about the mode-conditioned target state*. Using the principle of orthogonality [4], this is written as

$$E\left\{\begin{bmatrix}\omega_1(k)\\ \omega_2(k)\end{bmatrix}\tilde{\mathbf{x}}_{\mathrm{F}}^n(k)' \mid M(k)=n\right\} = \mathbf{0}. \tag{A2}$$

This holds because $\tilde{\mathbf{x}}_{\mathrm{F}}^n(k)$ is a linear combination of the MCEE $\tilde{\mathbf{x}}_j^m(k)$, $m = 1, 2$, $j = 1, 2$ according to equation (15). Because of Property 2 and this linear combination, $\tilde{\mathbf{x}}_{\mathrm{F}}^n(k)$ is also orthogonal to the LR.

Property 4: *The received MCE do not contain information about the target's mode probability.* This can be proved without the linear-Gaussian approximation using the fact that the likelihood of the mode based on the received MCE is a diffuse (noninformative) PDF; the state vector is composed of integrated (i.e., nonstationary) position and velocity:

$$p[\hat{\mathbf{x}}_1^1, \hat{\mathbf{x}}_1^2, \hat{\mathbf{x}}_2^1, \hat{\mathbf{x}}_2^2 \mid M=m] \to 0, \quad m = 1, 2, \tag{A3}$$

for any values of the MCE. Note that this property is not necessarily satisfied if the state vector contains stationary process(es), e.g., discretized Ornstein–Uhlembeck acceleration, autoregressive processes, or if the $\mathbf{F}$ matrix switches with the mode.

Property 5: *The components of the MCEE covariance matrices are orthogonal to the MCEEs and the LRs.* In any single-mode linear-Gaussian system, this is easily proved since the computed state covariance matrix does not depend on the observations. Let $q_j^{l(n,o)}(k) = q_j^{l(o,n)}(k)$ denote the locally computed scalar error covariance of the n-th and o-th state components (i.e., the n-th, o-th and the o-th, n-th components of $\mathbf{P}_j^l(k|k)$ as computed by the jth IMM estimator). This quantity is understood to be the covariance of the error of the l-th MCE conditioned on LT measurements up to k and target mode l being in effect. It can be stated as

$$q_j^{l(n,o)}(k) \triangleq E\left[\tilde{x}^{l(n)}(k)\tilde{x}(k)^{l(o)} \mid \mathbf{Z}_j^k, M(k)=l\right]. \tag{A4}$$

Conditioned on mode n at the FC, the first correlation to be analyzed is that of $q_j^{l(n,o)}(k)$ and any MCEE $\tilde{\mathbf{x}}_j^p(k)$. Since any zero mean Gaussian random variable is uncorrelated to the product of any other two Gaussian random variables,

$$\begin{aligned} &E\left[q_j^{l(n,o)}(k)\tilde{\mathbf{x}}_j^p(k) \mid M(k)=n\right] \\ &\quad = E\Bigg[E\left[\tilde{x}^{l(n)}(k)\tilde{x}^{l(n)}(k) \mid \mathbf{Z}_j^k, M(k)=l\right] \\ &\qquad \cdot \tilde{\mathbf{x}}_j^p(k) \mid M(k)=n\Bigg] \\ &\quad = \mathbf{0}, \end{aligned} \tag{A5}$$

and similar analysis can be used to also prove the lack of correlation between $q_j^{l(n,o)}(k)$ and $[\omega_1(k), \omega_2(k)]'$.

Property 6: *Given the LT MCEs, the received covariance components $q_j^{l(n,o)}(k)$ do not contain any linearly dependent information about the mode-conditioned target state $\mathbf{x}^{|n}(k)$*. This can be proved using the principle of orthogonality: the fused mode-conditioned target state estimate error $\tilde{\mathbf{x}}_{\mathrm{F}}^n(k)$ is a linear combination of the MCEE, which are uncorrelated to $q_j^{l(n,o)}(k)$ according to Property 5.

Property 7: *The received covariance components $q_j^{l(n,o)}(k)$ do not contain significant information about the target dynamic mode $M(k)$ or the mode-conditioned target state $\mathbf{x}^{|n}(k)$*. Because of the nonlinear operations involved in computing $\mathbf{P}_j^l(k|k)$, it is difficult to prove this. However, an approximate argument can be made based on the nature and intention of the IMM estimator. First, it is stated in Section IV-D that the MCC do not depend on the state or measurements at the current time step, so not much information should be expected. They do, however, depend on the previous MCE and mode probabilities because of the SOM matrix term. But the SOM is only a result of the mixing process that prevents the exponential growth of mode history hypotheses, and this method allows for feasible, yet suboptimal, multiple-model estimation. Certainly, in single-mode Gaussian

systems, or when using the infeasible optimal multiple-model estimator [4], covariance/crosscovariance matrices are deterministic, carry no statistical information, and can be easily computed at the FC. For example, this can be easily seen in the special case where the target is known to be in a specific state at the previous time step: there is no SOM in this case.

If system designers do not have access to LT design parameters or the target motion parameters, then it makes sense why received covariance matrices would be part of the fusion process. In such unfortunate scenarios, the FC must utilize a highly approximate fusion algorithm like covariance intersection or Chernoff fusion. However, knowledge of the computational pipeline of the LT track parameters allows for systematic, model-based Bayesian fusion as presented here.

Property 8: *Direct access to the locally-computed, mode-conditioned Kalman gains and innovation covariances does not improve the fused target state estimate or the consistency of the fused estimate covariance; likewise, computing and incorporating the covariance of these locally-computed covariance components does not affect the fused estimate or fused covariance*. This is a direct consequence of Property 5. It can be shown that because the components of the MCC are uncorrelated to the MCEE and the LR, then the MCEE and LR covariances are unaffected by the covariance of the LT Kalman gains and innovation covariance components. Only the mode-conditioned means of these matrices need to be computed at the FC as presented in Section IV-D.

REFERENCES

[1] D. Acar and U. Orguner
"Information decorrelation for an interacting multiple model filter,"
in *Proc. 21st Conf. on Inf. Fusion*, Jul. 2018.

[2] Y. Bar-Shalom
"On the track-to-track correlation problem,"
IEEE Trans. Automatic Control, vol. AC-26, no. 2, pp. 571–572, Apr. 1981.

[3] Y. Bar-Shalom and L. Campo
"The effect of the common process noise on the two-sensor fused-track covariance, "
IEEE Trans. Aerosp. Electron. Syst., vol. AES-22, no. 6, pp. 803–805, Nov. 1986.

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

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

[6] C. Chong, S. Mori, F. Govaers, and W. Koch
"Comparison of tracklet fusion and distributed Kalman filter for track fusion,"
in *Proc. 17th Int. Conf. on Inf. Fusion (FUSION)*, Jul. 2014, pp. 1–8.

[7] C. Y. Chong and S. Mori
"Graphical models for nonlinear distributed estimation,"
in *Proc. 7th Int. Conf. Inf. Fusion*, Jul. 2004.

[8] M. Günay, U. Orguner, and M. Demirekler
"Approximate Chernoff fusion of Gaussian mixtures using sigma-points,"
in *Proc. 17th Int. Conf. on Inf. Fusion (FUSION)*, Jul. 2014, pp. 870–877.

[9] M. Günay, U. Orguner, and M. Demirekler
"Chernoff fusion of Gaussian mixtures for distributed maneuvering target tracking,"
in *Proc. 18th Int. Conf. Inf. Fusion (FUSION)*, Jul. 2015, pp. 870–877.

[10] M. B. Hurley
"An information theoretic justification for covariance intersection and its generalization,"
in *Proc. 5th Int. Conf. Inf. Fusion (FUSION)*, Jul. 2002, pp. 505–511.

[11] L. M. Kaplan, W. D. Blair, and Y. Bar-Shalom
"Simulations studies of multisensor track association and fusion methods,"
in *Proc. 2006 IEEE Aerosp. Conf.*, 2006, pp. 16.

[12] K. Lu and R. Zhou
"Sensor fusion of Gaussian mixtures for ballistic target tracking in the re-entry phase,"
Sensors, vol. 16, no. 8, Aug. 2016.

[13] R. P. S. Mahler
"Optimal/robust distributed data fusion: a unified approach,"
Proc. SPIE, vol. 4052, 2000.

[14] B. Noack, M. Reinhardt, and U. D. Hanebeck
"On nonlinear track-to-track fusion with Gaussian mixtures,"
in *Proc. 17th Int. Conf. Inf. Fusion (FUSION)*, 2014, pp. 1–8.

[15] A. C. Rencher and G. B. Schaalje
Linear Models in Statistics, 2nd ed. Wiley, Stockholm, Sweden, 2008.

Radu Visina received the Ph.D. in electrical engineering from the University of Connecticut in 2019, where he contributed to systems engineering with a focus on radar target tracking. Before pursuing graduate studies, Dr Visina designed and developed high-performance, high-precision control systems and supporting software for the sensor calibration as well as power systems industries. Dr Visina maintains expertise in the sub-fields of nonlinear estimation and feedback control, maneuvering/multiple model target tracking, nonlinear information fusion, Bayesian decision theory, and urban target tracking using non-line-of-sight radar measurement extractions. Dr Visina is currently Technical Lead Research Engineer with the Information Systems Laboratories developing real-time RF simulation tools and terrain knowledge-aided radar target tracking techniques.

Yaakov Bar-Shalom (F'84) received the B.S. and M.S. degrees from the Technion in 1963 and 1967, respectively, and the Ph.D. degree from Princeton University in 1970, all in electrical engineering. Currently he is Board of Trustees Distinguished Professor in the Electrical and Computer Engineering Department and Marianne E. Klewin Professor with the University of Connecticut. His current research interests are in estimation theory, target tracking, and data fusion. He has published more than 600 papers and book chapters. He coauthored/edited eight books, including *Tracking and Data Fusion* (YBS Publishing, 2011). In 2002, he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. 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.

Peter Willett has been a faculty member in the Electrical and Computer Engineering Department at the University of Connecticut since 1986. Since 1998 he has been a Professor, and since 2003 an IEEE Fellow. He is Chief Editor for the IEEE Aerospace and Electronic Systems Magazine (2018–2020). He was Editor-in-Chief for the IEEE Signal Processing Letters, 2014–2016 and before that for the IEEE Transactions on Aerospace and Electronic Systems from 2006 to 2011. He was also the AESS Vice President for Publications from 2012 to 2014. He is a member of the IEEE Fellows Committee, Ethics Committee, and Periodicals Committee, and the IEEE Signal Processing Society's Technical Activities and Conference Boards. He is a member of the IEEE AESS Board of Governors and was Chair of IEEE Signal Processing Society's Sensor-Array and Multichannel (SAM) technical committee.

Dipak K. Dey is a Board of Trustees Distinguished Professor of Statistics with the University of Connecticut. He was the Department Head of Statistics, and later Associate Dean of Research and Physical Sciences in the College of Liberal Arts and Sciences at the University of Connecticut. He has published 10 books/edited volumes and more than 300 refereed journal articles and book chapters in various statistical and interdisciplinary journals. Currently, he is the Editor-in-Chief for *Sankhya, the Indian Journal of Statistics*. His research areas include Bayesian methodology and applications involving categorical and longitudinal data, clustering and classification, and spatio-temporal and survival data analysis.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

ISIF Website: http://www.isif.org

2020 BOARD OF DIRECTORS*

2018–2020	2019–2021	2020–2022
Fredrik Gustafsson	Kathryn Laskey	Pieter De Villiers
X. Rong Li	Felix Govaers	Murat Efe
Zhansheng Duan	Simon Maskell	Wolfgang Koch

*Board of Directors are elected by the members of ISIF for a three year term.

PAST PRESIDENTS

Paulo Costa, 2019	Joachim Biermann, 2011	Xiao-Rong Li, 2003
Lyudmila Mihaylova, 2018	Stefano Coraluppi, 2010	Yaakov Bar-Shalom, 2002
Lyudmila Mihaylova, 2017	Elisa Shahbazian, 2009	Pramod Varshney, 2001
Jean Dezert, 2016	Darko Musicki, 2008	Yaakov Bar-Shalom, 2000
Darin Dunham, 2015	Erik Blasch, 2007	Jim Llinas, 1999
Darin Dunham, 2014	Pierre Valin, 2006	Jim Llinas, 1998
Wolfgang Koch, 2013	W. Dale Blair, 2005	
Roy Streit, 2012	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.

Journal of Advances in Information Fusion Volume 15 Number 2 December 2020 ISSN 1557-6418