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.

1557-6418/20/$17.00 © 2020 JAIF

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 multiplying 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

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

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

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

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

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

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)

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

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

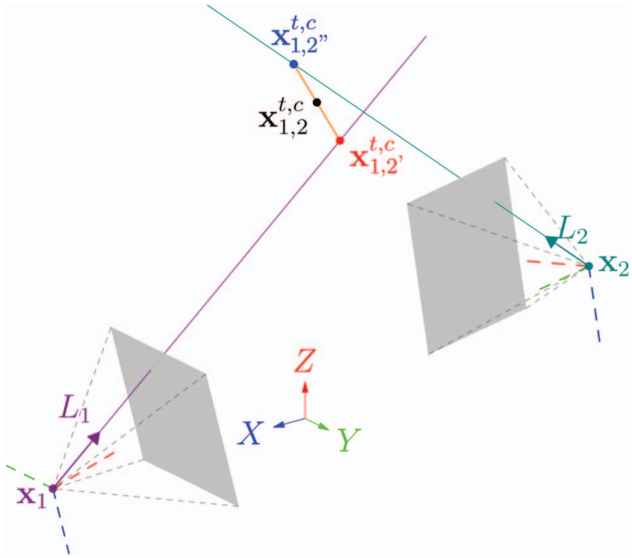

Fig. 1. Using CPA to convert azimuth measurements into 3-D Cartesian measurements. The sensors have their own rotated Cartesian frames with respect to the common Cartesian frame (ECI or ECEF) shown in the center. The LOS measurements are present as rays L_1 and L_2. The two closest positions on the LOSs 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

$$\zeta_{1,2}^{t,b,c}(k) \approx \begin{bmatrix} \boldsymbol{x}^t(k) \\ \boldsymbol{x}^t(k) \end{bmatrix} + \begin{bmatrix} \boldsymbol{w}_{12'}^{t,c}(k) \\ \boldsymbol{w}_{12''}^{t,c}(k) \end{bmatrix}$$

$$+B_{1,2}^t(k) \begin{bmatrix} C_1^{t,b}(k) & \boldsymbol{0} \\ \boldsymbol{0} & C_2^{t,b}(k) \end{bmatrix} \begin{bmatrix} \theta_1^b \\ \phi_1^b \\ \psi_1^b \\ \theta_2^b \\ \phi_2^b \\ \psi_2^b \end{bmatrix}$$

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

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

$$\boldsymbol{w}_{1,2}^{t,b,c}(k) = \begin{bmatrix} \boldsymbol{w}_{12'}^{t,c}(k) \\ \boldsymbol{w}_{12''}^{t,c}(k) \end{bmatrix}$$

$$\approx B_{1,2}^t(k) \begin{bmatrix} \boldsymbol{w}_1^{t,b}(k) \\ \boldsymbol{w}_2^{t,b}(k) \end{bmatrix}$$

$$\sim \mathcal{N}\left(\boldsymbol{0}_{10\times 1}, R_{1,2}^{t,b,c}(k)\right), \quad (25)$$

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

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

where

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

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

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

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

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

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

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,

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

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

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

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

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

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

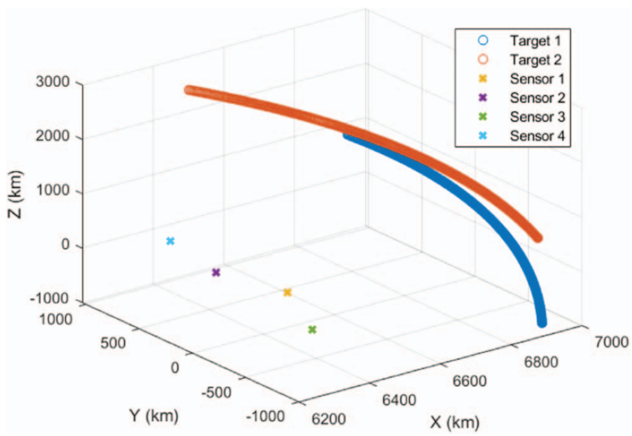

Fig. 2. Long-range sensor and target setup in ECEF coordinates. $K = 350$ s.

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.

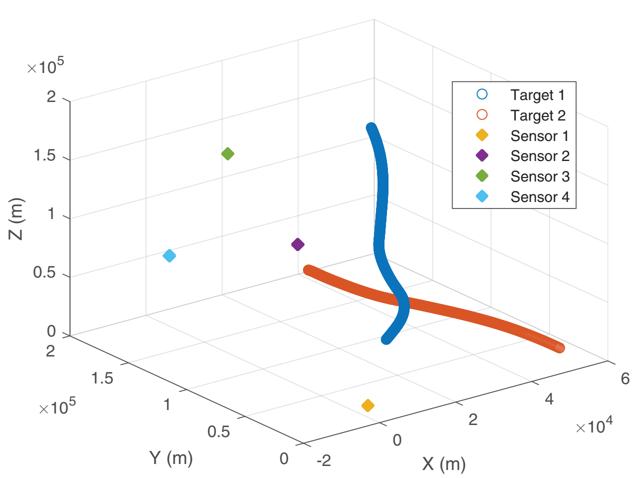

Fig. 3. Short-range sensor and target setup in ECEF coordinates. $K = 400$ s.

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, $n_{\mathrm{MC}} = 100$ 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, $n_{\mathrm{MC}} = 100$ 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		

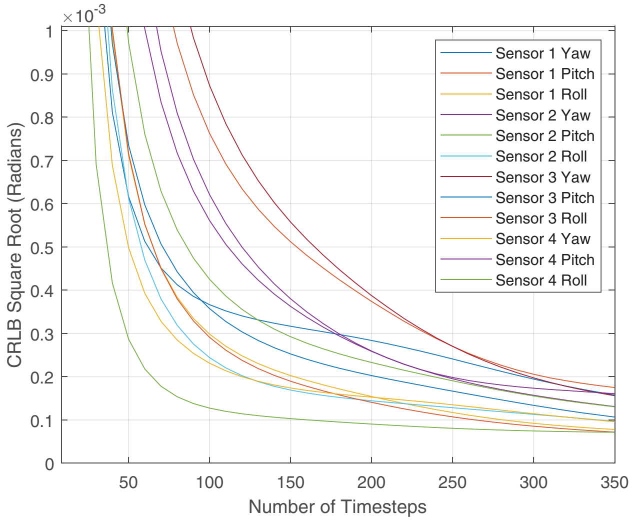

Fig. 4. CRLB square root of bias estimates compared to number of time steps for the short-range scenario with two targets.

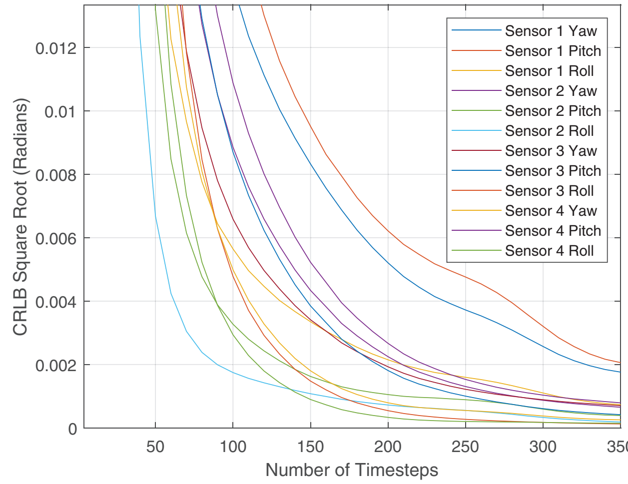

Fig. 6. CRLB square root of bias estimates compared to number of time steps for the short-range scenario with one target.

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.

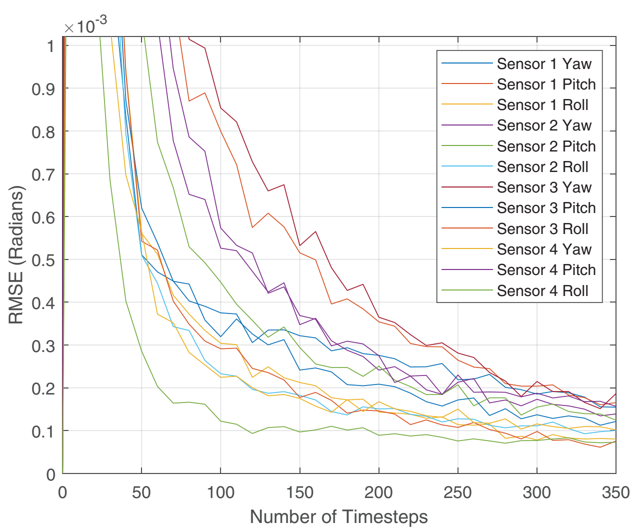

Fig. 5. RMSE of bias estimates compared to number of time steps for the short-range scenario with two targets.

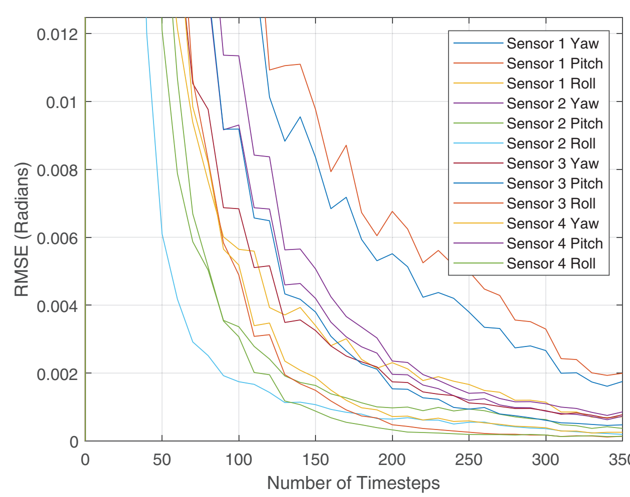

Fig. 7. CRLB square root of bias estimates compared to number of time steps for the short-range scenario with one target.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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.