Unbiased Conversion of Passive Sensor Measurements Using Closest Point of Approach

MICHAEL KOWALSKI
YAAKOV BAR-SHALOM
PETER WILLETT
TIM FAIR

In passive sensor target tracking, there are applications that require converting the angle-only measurements into Cartesian space. Multisensor methods can be used to convert the raw measurements into Cartesian measurements by finding the intersection of lines of sight. This method contains significant nonlinearity in its conversion; therefore, it is subject to corresponding errors such as a conversion bias and an improper covariance. The proposed method uses a second-order Taylor series expansion to accurately account for the conversion nonlinearities. This results in an explicit (noniterative) expression of the Cartesian position based on two line-of-sight measurements in three dimensions. This paper investigates the severity of the conversion biases from nonlinearity and the efficiency of the unbiased conversion with regard to compensating for them.

Manuscript received October 31, 2021; revised November 27, 2021; released for publication July 7, 2022.

Associate Editor: Florian Meyer.

M. Kowalski, Y. Bar-Shalom, and P. Willett are with the Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269 USA (e-mail: michael.p.kowalski@uconn.edu; ybs@uconn.edu; peter.willett@uconn.edu).

T. Fair is with the Toyon Research Corporation, Sterling, VA 93117 USA (e-mail: tfair@toyon.com).

1557-6418/22/$17.00

I. INTRODUCTION

Passive sensors are especially challenging compared to those that are active. Although some passive sensors also deliver amplitude information—and, indeed, in some situations, can incorporate processing such as via wavefront curvature or based on target image expectations to infer range—they are commonly assumed to present only angular measurements (see [3], [4]). Passive sensor measurements require a data fusion step if a three-dimensional plot is desired, as it often is by a downstream tracker. However, as we shall see, a typical processing chain aimed at such plots produces an unwanted but modelable bias.[1] This paper addresses such modeling, but we note that while there are sophisticated ways to mitigate bias (e.g., nonlinear least-squares in [2] and maximum-likelihood (ML) estimation in [13]), the goal here is for a simple delivery of Cartesian measurements to a simple Cartesian tracker such as a Kalman filter; a nonlinear dynamic estimation approach such as a particle filter does not require such pre-processing, but is often much more computationally demanding. However, in any case, even therein improvements can be made in terms of initialization (using the angle measurements to obtain Cartesian positions and velocities), and sensor registration through bias estimation is a widely researched topic that commonly uses converted measurements from spherical coordinates and range/direction-sines coordinates [16].

Triangulation is often used to convert two angle-only measurements into Cartesian position. In [14], triangulation is used to initialize an ML approach to converting angle-only measurements into Cartesian. However, the ML method uses a search to obtain an (iterative) estimate of the Cartesian position. In the case of an ML search, it is practically impossible to produce a Jacobian matrix of the converted Cartesian coordinates with respect to the original angle-only measurements as the "location" of the ML estimate's convergence point is not analytically related to the angle-only measurements. This Jacobian is useful for applications in other target tracking applications such as bias estimation. The novel [10] was the first to attempt the second-order Taylor series expansion for angle-only measurements in passive sensors and use them for estimation of sensor bias, which is separate from conversion bias. The work relied on triangulation to form bias pseudo-measurements. Conversion via triangulation is flawed particularly as the conversion is overly reliant on azimuth rather than the total angle. The transformation fails when the lines of sight of the two sensors are equal in azimuth as the denominator of the conversion equation becomes zero. This would result in "blind spots" of the transformation where the error and covariance consistency would dete-

[1] Other effects such as refraction also produce offsets that might be called biases, but these can be treated in other ways and we do not discuss them here.

riorate greatly, as seen in [9]. The law of sines is used in [5] to generate ranges in order to convert the angle measurements into Cartesian similarly to triangulation. In [15], the closest point of approach between the two line-of-sight (LOS) rays is used to create composite Cartesian coordinates. This paper investigates the closest point of approach method to produce a (noniterative) expression of the Cartesian position based on two LOS measurements in three dimensions. However, the conversion of angle measurements into Cartesian is a nonlinear transformation that requires an unbiased conversion [3]. As such, this paper investigates the bias of the explicit conversion expression and provides a solution to overcome it.

An approach to deriving an unbiased conversion for a nonlinear transformation is used in [17] to convert sine space coordinates into Cartesian by using a second-order Taylor series expansion. In this paper, the approach is replicated for the conversion of passive sensor measurements to Cartesian. The method is also similarly evaluated to ensure that the second-order conversion is necessary.

In particular, it is necessary to account for the nonlinear measurement conversion by accounting for the bias and also by converting the measurement noise covariance. The conversion bias is well documented in the literature [3], [12] for spherical to Cartesian conversion where the noise pattern resembles a curved lens in Cartesian space rather than a sphere. It is necessary to adjust the converted noise covariance to more accurately represent the converted measurements. Similarly, the same process must be made for the conversion of angle-only measurements into Cartesian. Previously, in [9], the conversion bias and covariance consistency errors were improved by using the second-order Taylor series expansion. A simulation was constructed that produced a conversion bias, and the unbiased conversion reduced it to less than one-tenth of its original value. This conversion was then used in [10] to form bias pseudo-measurements, which could then be used to estimate sensor biases without needing to estimate the target state. In [11], debiased polar measurements were used in a Kalman Filter and are shown to improve tracking performance relative to a mixed coordinate Extended Kalman Filter (EKF). The RMS values were reduced as a result of the reduction of errors due to nonlinearity. This is because an EKF is subject to the same nonlinearities when converting its state to sensor coordinates when using a mixed measurement model, albeit in the opposite "direction".

A cubature integration method can also be used to approximate the moments of the nonlinear conversion. In [6], several filters involving cubature methods were examined with respect to areas of severe nonlinearity. The cubature Kalman filters and a Kalman filter using converted measurements were found to be commensurate in performance when wrapping was used.

This paper is outlined as follows: Section II contains the definitions for passive sensing used by this method. The proposed method of closest point of approach is presented in Section III. The methods of analyzing the conversion are in Section IV. The parameters for the simulation are included in Section IV-A. The methods include analysis of the bias discussed in Section IV-B and analysis of the covariance discussed in Section IV-C. Section V concludes the paper.

Notation used in this work includes vectors as bold symbols such as $\mathbf{x}$. Gradients are defined using the ∇ symbol. The superscript t and subscript s specify the target and sensor indices, respectively. The superscripts c, db, and m specify that a variable is converted, debiased, and measured, respectively. The notation $'$ means the vector or matrix is transposed. The subscripts x, y, and z are used to specify that a variable refers to the respective Cartesian coordinate. The notation (k) specifies the time index. $\mathcal{N}$ specifies a Gaussian random variable with the mean and covariance in parentheses.

II. PROBLEM FORMULATION

Passive three-dimensional angle-only sensors are used for this paper. Passive sensors only give angle measurements pointing in the direction of the target. In three dimensions, this is made up of two angles, azimuth, and elevation. The sensors are assumed to be synchronous and the network consists of N_s sensors. At a timestep k, the position of sensor s in Cartesian space, assumed to be known, is

$$\mathbf{x}_s(k) = [x_s(k),\ y_s(k),\ z_s(k)]'. \tag{1}$$

For simplicity, there is only a single target t, and its Cartesian position is similarly

$$\mathbf{x}^t(k) = [x^t(k),\ y^t(k),\ z^t(k)]'. \tag{2}$$

The sensors generate measurements of the target from their own reference frame:

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

Using the positions derived in (3), the sensors generate elevation and azimuth measurements. Azimuth is denoted as α and elevation is denoted as ϵ. The measurements use atan2, which (MATLAB notation) is the four-quadrant inverse tangent.

$$\boldsymbol{\xi}_s(k) = \begin{bmatrix} \alpha_s(k) \\ \epsilon_s(k) \end{bmatrix} = \begin{bmatrix} \text{atan2}\left(\frac{y_s^t(k)}{x_s^t(k)}\right) \\ \text{atan2}\left(\frac{z_s^t(k)}{\sqrt{x_s^t(k)^2 + y_s^t(k)^2}}\right) \end{bmatrix}. \tag{4}$$

The measurements are combined with noises $w_s^\alpha(k)$ and $w_s^\epsilon(k)$ to produce the final measurement model, where superscript m signifies that the angles are the measurements generated by the sensors. The noise for each sensor is assumed uncorrelated independent white

Gaussian with variances $(\sigma_s^\alpha)^2$ and $(\sigma_s^\epsilon)^2$ and zero mean.[2] The noisy measurements are

$$\boldsymbol{\xi}_s^m(k) = \begin{bmatrix} \alpha_s^m(k) \\ \epsilon_s^m(k) \end{bmatrix} = \begin{bmatrix} \alpha_s(k) \\ \epsilon_s(k) \end{bmatrix} + \begin{bmatrix} w_s^\alpha(k) \\ w_s^\epsilon(k) \end{bmatrix}, \quad (5)$$

$$w_s^\alpha(k) \sim \mathcal{N}(0, (\sigma_s^\alpha)^2) \quad w_s^\epsilon(k) \sim \mathcal{N}(0, (\sigma_s^\epsilon)^2). \quad (6)$$

The noise variances are assumed to be known by the system. These measurements are assumed to be synchronous, but this model can be modified for the asynchronous case. The true target state is treated as a parameter because we consider a single point in time, and the method proposed here seeks to be agnostic with regard to target dynamics. A filter (which operates across time) normally introduces process noise models for tracking, but our method converts the individual measurements to be used in a tracking filter with the converted noise errors being zero-mean, with consistent covariance, and independent across time. As such, this method can be used with any target motion or process noise model.

In this paper, we assume sensors deliver synchronized measurements. In practice, a high frame rate mitigates asynchronicity. But, it is admitted that significantly asynchronous measurements will cause additional errors as they would need to be propagated to the same time for conversion, and this would require integration of bias estimation into the dynamic estimation procedure. That is beyond the scope of this paper, but is an intriguing topic for future work.

III. CONVERSION USING CLOSEST POINT OF APPROACH

A. The Conversion

The method for conversion investigated in this work is using the closest point of approach to generate a single Cartesian measurement using angle measurements from two sensors. This method finds the points on the two LOS rays that are closest to each other and then labels the midpoint between the two points as the composite Cartesian position. When converting lines of sight into Cartesian, it is important to take into account observability. Previously, in [9], the triangulation conversion was examined, but this was found to have observability problems when solely the azimuth components were similar, regardless of whether the lines of sight were parallel or not. This reduces the practicality of the approach because the reference frame would have to be rotated to avoid having the azimuth components be the same. The conversion proposed in this work is an improvement because it avoids this severe reliance on one of the two angle measurements, improving observability and avoiding the need for rotation. This process is shown in Fig. 1. The superscript c is used to signify that it is a conversion:

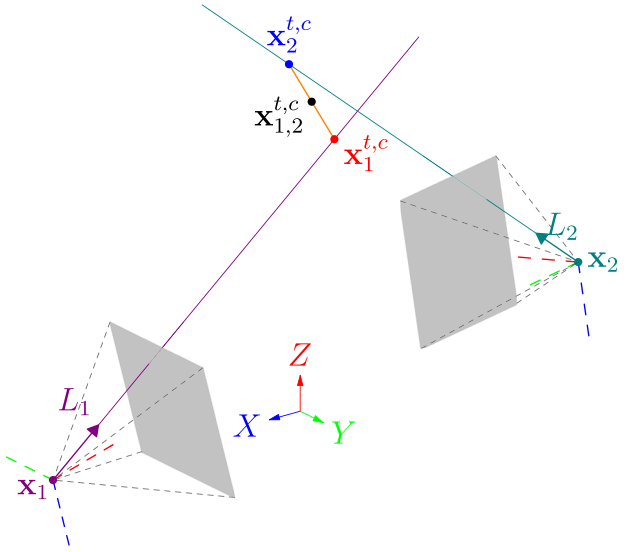


Fig. 1. Using CPA (closest point of approach) to convert azimuth measurements into three-dimensional Cartesian measurements.

$$\mathbf{x}_1^{t,c}(k) = \begin{bmatrix} x_1^{t,c}(k) \\ y_1^{t,c}(k) \\ z_1^{t,c}(k) \end{bmatrix} = \mathbf{x}_1(k) + \frac{\mathbf{L}_1(k)(\mathbf{L}_1(k)'\mathbf{B}_{1,2}(k)) - (\mathbf{L}_1(k)'\mathbf{L}_2(k))(\mathbf{L}_2(k)'\mathbf{B}_{1,2}(k))}{1-(\mathbf{L}_1(k)'\mathbf{L}_2(k))^2}, \quad (7)$$

$$\mathbf{x}_2^{t,c}(k) = \begin{bmatrix} x_2^{t,c}(k) \\ y_2^{t,c}(k) \\ z_2^{t,c}(k) \end{bmatrix} = \mathbf{x}_2(k) + \frac{\mathbf{L}_2(k)(\mathbf{L}_1(k)'\mathbf{L}_2(k))(\mathbf{L}_1(k)'\mathbf{B}_{1,2}(k)) - (\mathbf{L}_2(k)'\mathbf{B}_{1,2}(k))}{1-(\mathbf{L}_1(k)'\mathbf{L}_2(k))^2}, \quad (8)$$

$$\mathbf{x}_{1,2}^{t,c}(k) = \tfrac{1}{2}\left(\mathbf{x}_1^{t,c}(k) + \mathbf{x}_2^{t,c}(k)\right). \quad (9)$$

The conversion relies on the Cartesian vectors of the lines of sight defined as $\mathbf{L}$ and the line between the sensors defined as $\mathbf{B}$. These are calculated as

$$\mathbf{L}_1(k) = \begin{bmatrix} \cos(\alpha_1(k))\cos(\epsilon_1(k)) \\ \sin(\alpha_1(k))\cos(\epsilon_1(k)) \\ \sin(\epsilon_1(k)) \end{bmatrix}, \quad (10)$$

$$\mathbf{L}_2(k) = \begin{bmatrix} \cos(\alpha_2(k))\cos(\epsilon_2(k)) \\ \sin(\alpha_2(k))\cos(\epsilon_2(k)) \\ \sin(\epsilon_2(k)) \end{bmatrix}, \quad (11)$$

[2] In passive sensor angle measurements, the noise is commonly approximated with a Gaussian, but it is not the most accurate model for real sensors. Research in passive sensors has used alternatives such as the Kent distribution in [8] or a wrapped distribution as in [6]. For the purposes of this work, the noise is approximated as Gaussian. A Gaussian assumption yields a simple algorithm, so we use it; but, if a higher fidelity/complexity solution is required, the approaches in [10] and [4] should be consulted.

$$\mathbf{B}_{1,2}(k) = \mathbf{x}_2(k) - \mathbf{x}_1(k) = \begin{bmatrix} x_2(k) - x_1(k) \\ y_2(k) - y_1(k) \\ z_2(k) - z_1(k) \end{bmatrix}. \quad (12)$$

The derivation for these equations is included in the Appendix. In ML applications [10], the two Cartesian coordinate measurements can be used separately rather than merged for the parameter estimate.

In the presence of noise, this method is imperfect as the two lines of sight will not intersect exactly in three-dimensional space. As such, estimation methods are commonly used to estimate the true target state, which will be discussed later in the paper. However, in circumstances when estimation methods are undesirable, it is possible to use this method, and the effect of the noise can be approximated with a conversion. Furthermore, the noise can cause a bias in the converted measurement, but this bias can be calculated and removed.

It is important to note that this is an explicit expression of the conversion. Unlike iterative methods such as ML, it is possible to calculate a Jacobian of the converted Cartesian positions with respect to the original LOS measurements. As such, the derivatives can be calculated and, in turn, used to calculate the bias and covariance. The explicit expression and these derivatives are useful in other applications such as the generation of pseudo-measurements for bias estimation [10].

B. The Bias of the Conversion and Its Compensation

To approximate the noise covariance and debias the converted measurements, a Taylor series expansion is used. A second-order expansion is used here; however, further orders can be used for a more accurate conversion. For simplicity, the y and z expansions as well as the individual derivatives are moved to the Appendix. The superscript m denotes that the converted Cartesian measurements are made with the noisy LOS measurements:

$$\begin{aligned} x_{1,2}^{t,c,m}(k) \approx\ & x^t(k) + \frac{\partial x_{1,2}^{t,c}}{\partial \alpha_1} w_1^{\alpha}(k) + \frac{\partial x_{1,2}^{t,c}}{\partial \alpha_2} w_2^{\alpha}(k) \\ & + \frac{\partial x_{1,2}^{t,c}}{\partial \epsilon_1} w_1^{\epsilon}(k) + \frac{\partial x_{1,2}^{t,c}}{\partial \epsilon_2} w_2^{\epsilon}(k) \\ & + 0.5 \frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_1^2} w_1^{\alpha}(k)^2 + 0.5 \frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_2^2} w_2^{\alpha}(k)^2 \\ & + 0.5 \frac{\partial^2 x_{1,2}^{t,c}}{\partial \epsilon_1^2} w_1^{\epsilon}(k)^2 + 0.5 \frac{\partial^2 x_{1,2}^{t,c}}{\partial \epsilon_2^2} w_2^{\epsilon}(k)^2 \\ & + \frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_1 \partial \alpha_2} w_1^{\alpha}(k) w_2^{\alpha}(k) + \frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_1} w_1^{\alpha}(k) w_1^{\epsilon}(k) \\ & + \frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_2} w_1^{\alpha}(k) w_2^{\epsilon}(k) + \frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_1} w_2^{\alpha}(k) w_1^{\epsilon}(k) \\ & + \frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_2} w_2^{\alpha}(k) w_2^{\epsilon}(k) + \frac{\partial^2 x_{1,2}^{t,c}}{\partial \epsilon_1 \partial \epsilon_2} w_1^{\epsilon}(k) w_2^{\epsilon}(k). \end{aligned} \quad (13)$$

The expected value of the expanded term contains the conversion bias.

$$\begin{aligned} E[x_{1,2}^{t,c,m}(k)] \approx\ & x^t(k) + 0.5 \frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_1^2} (\sigma_1^{\alpha})^2 \\ & + 0.5 \frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_2^2} (\sigma_2^{\alpha})^2 + 0.5 \frac{\partial^2 x_{1,2}^{t,c}}{\partial \epsilon_1^2} (\sigma_1^{\epsilon})^2 \\ & + 0.5 \frac{\partial^2 x_{1,2}^{t,c}}{\partial \epsilon_2^2} (\sigma_2^{\epsilon})^2, \end{aligned} \quad (14)$$

$$\begin{aligned} c_{x,1,2} =\ & 0.5 \frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_1^2} (\sigma_1^{\alpha})^2 + 0.5 \frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_2^2} (\sigma_2^{\alpha})^2 \\ & + 0.5 \frac{\partial^2 x_{1,2}^{t,c}}{\partial \epsilon_1^2} (\sigma_1^{\epsilon})^2 + 0.5 \frac{\partial^2 x_{1,2}^{t,c}}{\partial \epsilon_2^2} (\sigma_2^{\epsilon})^2, \end{aligned} \quad (15)$$

$$E[x_{1,2}^{t,c,m}(k)] \approx x^t(k) + c_{x,1,2}. \quad (16)$$

By calculating and subtracting the bias, it is possible to avoid this error, producing the debiased measurements denoted by the superscript db.

$$x_{1,2}^{t,c,m,db}(k) = x_{1,2}^{t,c,m}(k) - c_{x,1,2}, \quad (17)$$

$$E[x_{1,2}^{t,c,m,db}(k)] \approx x^t(k). \quad (18)$$

The converted state is rewritten into a simple form that contains the truth and converted zero-mean Gaussian noise. This results in the following measurement equation for the converted measurements:

$$\begin{aligned} \mathbf{x}_{1,2}^{t,c,m,db}(k) &= \begin{bmatrix} x_{1,2}^{t,c,m,db}(k) \\ y_{1,2}^{t,c,m,db}(k) \\ z_{1,2}^{t,c,m,db}(k) \end{bmatrix} \\ &= \begin{bmatrix} x^t(k) \\ y^t(k) \\ z^t(k) \end{bmatrix} + \begin{bmatrix} w_{1,2}^{x,t,c,db}(k) \\ w_{1,2}^{y,t,c,db}(k) \\ w_{1,2}^{z,t,c,db}(k) \end{bmatrix}. \end{aligned} \quad (19)$$

The noise after conversion is now zero-mean white Gaussian, as is desirable for tracking and applications,

$$\mathbf{w}_{1,2}(k) = \begin{bmatrix} w_{1,2}^{x,t,c,db}(k) \\ w_{1,2}^{y,t,c,db}(k) \\ w_{1,2}^{z,t,c,db}(k) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{R}_{1,2}^{t,c,db} \right). \quad (20)$$

C. The Covariance of the Converted Errors

The covariance matrix for the converted measurements can be calculated with the same Taylor series expansion. The full derivation of the following equations and the appropriate derivatives are found in the Appendix. The covariance matrix (with the superscripts

and subscripts removed for simplicity) is

$$\mathbf{R}_{1,2}^{t,c,db} = \begin{bmatrix} V(x(k)) & CV(x(k),y(k)) & CV(x(k),z(k)) \\ CV(x(k),y(k)) & V(y(k)) & CV(y(k),z(k)) \\ CV(x(k),z(k)) & CV(y(k),z(k)) & V(z(k)) \end{bmatrix}, \tag{21}$$

where V is the variance of the variable defined by

$$V(x_{1,2}^{t,c,m,db}(k)) = E[x_{1,2}^{t,c,m,db}(k)^2] - E[x_{1,2}^{t,c,m,db}(k)]^2. \tag{22}$$

By using the terms in the Taylor series expansion, the equation is changed into a usable form expressed as

$$\begin{aligned} &V(x_{1,2}^{t,c,m,db}(k)) = \\ &\left(\frac{\partial x_{1,2}^{t,c}}{\partial \alpha_1}\right)^2 (\sigma_1^\alpha)^2 + \left(\frac{\partial x_{1,2}^{t,c}}{\partial \alpha_2}\right)^2 (\sigma_2^\alpha)^2 \\ &+ \left(\frac{\partial x_{1,2}^{t,c}}{\partial \epsilon_1}\right)^2 (\sigma_1^\epsilon)^2 + \left(\frac{\partial x_{1,2}^{t,c}}{\partial \epsilon_2}\right)^2 (\sigma_2^\epsilon)^2 \\ &+ \left(\frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_1 \partial \alpha_2}\right)^2 (\sigma_1^\alpha)^2(\sigma_2^\alpha)^2 + \left(\frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_1}\right)^2 (\sigma_1^\alpha)^2(\sigma_1^\epsilon)^2 \\ &+ \left(\frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_2}\right)^2 (\sigma_1^\alpha)^2(\sigma_2^\epsilon)^2 + \left(\frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_1}\right)^2 (\sigma_2^\alpha)^2(\sigma_1^\epsilon)^2 \\ &+ \left(\frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_2}\right)^2 (\sigma_2^\alpha)^2(\sigma_2^\epsilon)^2 + \left(\frac{\partial^2 x_{1,2}^{t,c}}{\partial \epsilon_1 \partial \epsilon_2}\right)^2 (\sigma_1^\epsilon)^2(\sigma_2^\epsilon)^2, \end{aligned} \tag{23}$$

and the covariance CV is defined by

$$\begin{aligned} &CV(x_{1,2}^{t,c,m,db}(k), y_{1,2}^{t,c,m,db}(k)) \\ &\quad = E[(x_{1,2}^{t,c,m,db}(k) - E[x_{1,2}^{t,c,m,db}(k)]) \\ &\qquad \times (y_{1,2}^{t,c,m,db}(k) - E[y_{1,2}^{t,c,m,db}(k)])], \end{aligned} \tag{24}$$

where, similarly, the Taylor series expansion transforms the equation into

$$\begin{aligned} &\mathrm{Cov}(x_{1,2}^{t,c,m,db}(k), y_{1,2}^{t,c,m,db}(k)) \\ &= \left(\frac{\partial x_{1,2}^{t,c}}{\partial \alpha_1}\right)\left(\frac{\partial y_{1,2}^{t,c}}{\partial \alpha_1}\right)(\sigma_1^\alpha)^2 + \left(\frac{\partial x_{1,2}^{t,c}}{\partial \alpha_2}\right)\left(\frac{\partial y_{1,2}^{t,c}}{\partial \alpha_2}\right)(\sigma_2^\alpha)^2 \\ &\quad + \left(\frac{\partial x_{1,2}^{t,c}}{\partial \epsilon_1}\right)\left(\frac{\partial y_{1,2}^{t,c}}{\partial \epsilon_1}\right)(\sigma_1^\epsilon)^2 + \left(\frac{\partial x_{1,2}^{t,c}}{\partial \epsilon_2}\right)\left(\frac{\partial y_{1,2}^{t,c}}{\partial \epsilon_2}\right)(\sigma_2^\epsilon)^2 \\ &\quad + \left(\frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_1 \partial \alpha_2}\right)\left(\frac{\partial^2 y_{1,2}^{t,c}}{\partial \alpha_1 \partial \alpha_2}\right)(\sigma_1^\alpha)^2(\sigma_2^\alpha)^2 \\ &\quad + \left(\frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_1}\right)\left(\frac{\partial^2 y_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_1}\right)(\sigma_1^\alpha)^2(\sigma_1^\epsilon)^2 \\ &\quad + \left(\frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_2}\right)\left(\frac{\partial^2 y_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_2}\right)(\sigma_1^\alpha)^2(\sigma_2^\epsilon)^2 \\ &\quad + \left(\frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_1}\right)\left(\frac{\partial^2 y_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_1}\right)(\sigma_2^\alpha)^2(\sigma_1^\epsilon)^2 \\ &\quad + \left(\frac{\partial^2 x_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_2}\right)\left(\frac{\partial^2 y_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_2}\right)(\sigma_2^\alpha)^2(\sigma_2^\epsilon)^2 \\ &\quad + \left(\frac{\partial^2 x_{1,2}^{t,c}}{\partial \epsilon_1 \partial \epsilon_2}\right)\left(\frac{\partial^2 y_{1,2}^{t,c}}{\partial \epsilon_1 \partial \epsilon_2}\right)(\sigma_1^\epsilon)^2(\sigma_2^\epsilon)^2. \end{aligned} \tag{25}$$

At this point, the previous unbiased conversion and covariance evaluation can be defined as the second-order conversion. In the simulations, this will be compared to a first-order conversion, in which the same conversion via the closest point of approach is made, but debiasing is not performed and the second-order components are ignored in the covariance calculation.

IV. SIMULATIONS AND RESULTS

A. Simulation Parameters

To analyze the conversion, we study a long-range orbital scenario in which an orbiting target passes through the field of view of two sea-level sensors. The target starts at 7000 km from the center of the earth, or at an altitude of 622 km above sea level on the equator directly on the x axis in ECI (Earth-centered inertial) coordinates. The target begins with a velocity necessary for maintaining an orbit, 7.546 km/s, with the vector pointing at an angle of 2.678 radians (clockwise from the Y axis) in the Y–Z plane of ECI. Both sensors are stationary at sea level. Sensor 1 is at -1° in latitude and -3° in longitude, and sensor 2 is at 1° in latitude and 3° in longitude. The sensors move via the rotation of the earth with respect to ECI coordinates. The sensors are oriented facing directly up, meaning that the local vertical is the boresight and 0 azimuth position. This scenario is designed to have a target that passes the sensors, causing angle measurements that begin relatively perpendicular and become parallel over time. A less observable system is present when the measurements are parallel. Less observable systems benefit more from improved nonlinear conversion. The simulation setup is shown in Fig. 2.

The measurement noise values and simulation parameters are given in Table I and are kept the same for both sensors for simplicity. The performance of the conversion is analyzed over 100 000 Monte Carlo runs.

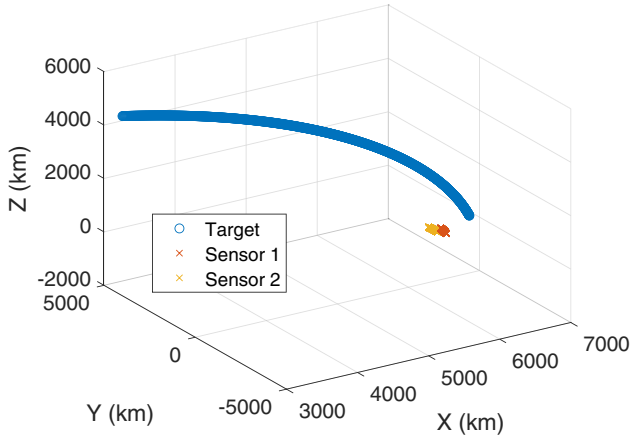


Fig. 2. Sensor and target setup in ECI.

B. Cartesian Position Bias Evaluation

The value of this conversion is first investigated by evaluating the significance of the bias. The conventional conversion is the first-order conversion, which is based on the first-order Taylor series expansion. The bias in the conversion is defined as

$$\mu_x = x^t(k) - E[x^{t,c,m}_{1,2}(k)], \tag{26}$$

$$\mu_y = y^t(k) - E[y^{t,c,m}_{1,2}(k)], \tag{27}$$

$$\mu_z = z^t(k) - E[z^{t,c,m}_{1,2}(k)]. \tag{28}$$

The significance of the bias is defined as the norm of the bias divided by the standard deviation of the noise in Cartesian coordinates. This metric is based on the metric proposed in [11] and used in [7]. The noise is roughly converted by multiplying the range from the closest sensor by the sine of the standard deviation of the azimuth,

$$\beta = \frac{\sqrt{\mu_x^2 + \mu_y^2 + \mu_z^2}}{\sin(\sigma_s^\alpha)\sqrt{(x_s^t(k))^2 + (y_s^t(k))^2 + (z_s^t(k))^2}}. \tag{29}$$

The biases and their significance over the simulation time steps are seen in Fig. 3. The bias increases significantly as the measurements become more parallel. The significance of the bias is quite low, but for long-distance applications, it is advantageous to include debiasing. If the bias significance is less than 0.3 in most applications, it is considered to be negligible as it is less than a 10% in-

Table I
Simulation Parameters, $N_s = 2, N_t = 1, K = 1000$ s, and $N_{\text{MC}} = 100\,000$ runs

	Sensor measurement noise	
	Azimuth noise standard deviation	Elevation noise standard deviation
Sensor 1	1 mrad	1 mrad
Sensor 2	1 mrad	1 mrad

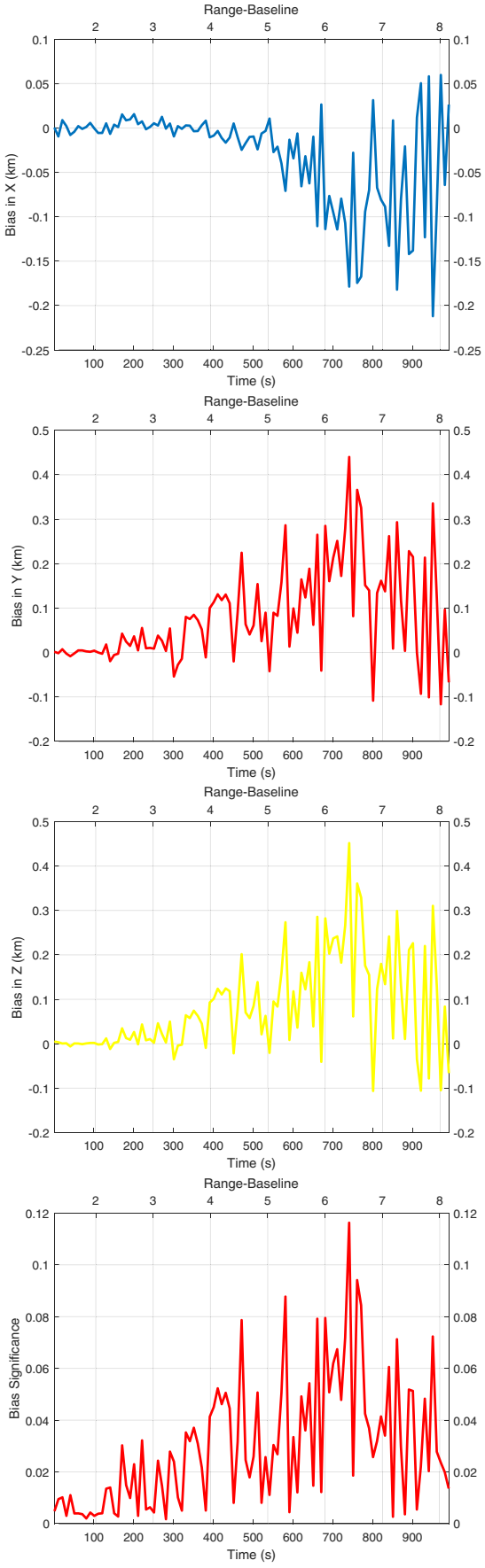


Fig. 3. Conversion bias and its significance over time.

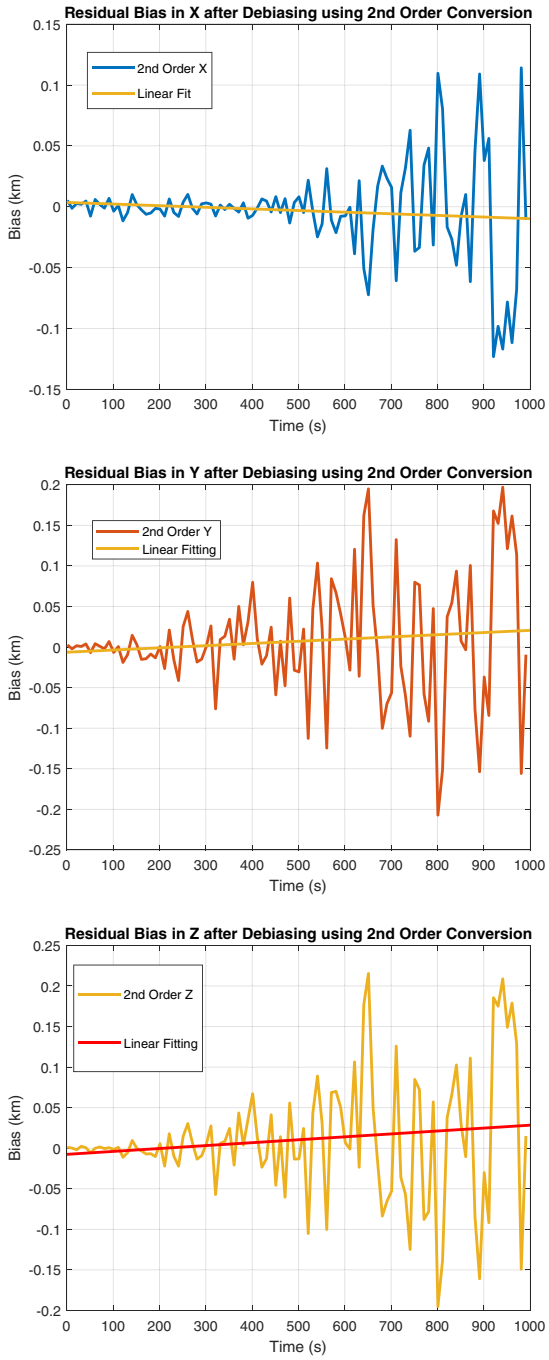


Fig. 4. Results of debiasing using the second-order conversion.

crease in the mean square error. The results of debiasing are seen in Fig. 4.

After debiasing, the mean of the results is appropriately centered on zero, meaning a significant improvement in measurement consistency is made. Higher order Taylor expansions may lead to more accurate debiasing.

C. Covariance Analysis

The accuracy of the covariance matrix from equation (21) is also analyzed for the second-order conversion. A Monte Carlo simulation is made to achieve this. For each

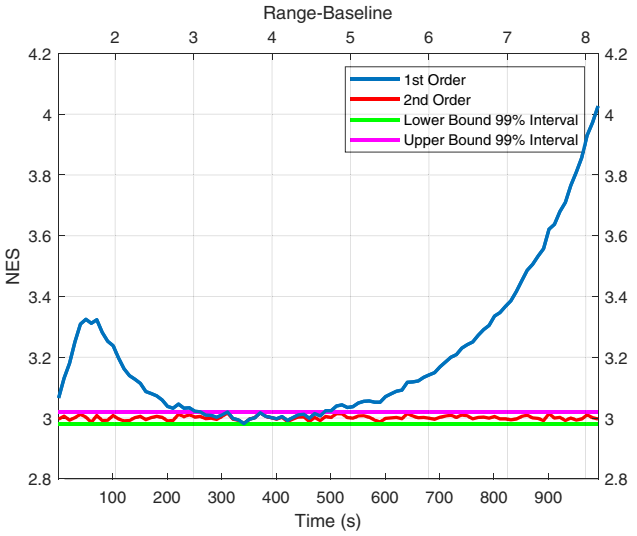


Fig. 5. NES First and Second Order Comparison of covariance NES.

Monte Carlo run, denoted by superscript n out of N_{MC}, a converted measurement is obtained and compared to the truth. The metric for analyzing the covariance is the Normalized Error Squared (NES). This is defined as

$$\tilde{\mathbf{x}}^t_{1,2}(k) = \mathbf{x}^{t,c,m,db,n}_{1,2}(k) - \mathbf{x}^t(k), \tag{30}$$

$$\mathrm{NES}(k) = \tfrac{1}{N_{\mathrm{MC}}} \textstyle\sum_{n=1}^{N_{\mathrm{MC}}} \tilde{\mathbf{x}}^t_{1,2}(k)' \mathbf{R}^{t,c,db,n}_{1,2}(k)^{-1} \tilde{\mathbf{x}}^t_{1,2}(k). \tag{31}$$

Comparisons of the NESs, both for the first- and for the second-order conversions to each other and to the 99% confidence region, are shown in Fig. 5. The NES for the first-order conversion is significantly higher than the confidence interval, meaning that the covariance calculated is not accurately containing the measurement points. In this case, the covariance is too small. The second-order conversion very accurately remedies this problem and results in a consistent NES. This calculated covariance matrix can be safely used to represent the converted Cartesian measurements.

D. Comparison With ML Conversion

In the previous sections, the proposed method is shown to remove nearly all bias and calculates an accurate, albeit pessimistic, covariance matrix. However, it is important to consider comparing the method with the already present method of generating composite measurements using ML. This method is presented in [14] and involves implementation of the ML using Iterated Least-Squares (ILS). Intuitively, one can deduce that the ML method should produce more accurate results for Cartesian coordinates from fusing two angle-only measurements because it is efficient compared to the Cramér–Rao Lower Bound (CRLB). However, the proposed method has an advantage in that it is an explicit (noniterative) expression of the Cartesian posi-

Table II
Comparison of Computation Times

	Proposed explicit conversion method	ML one-iteration conversion method	ML ten-iterations conversion method
$K = 100$ s	0.0084 s	0.0165 s	0.0966 s
$K = 1000$ s	0.0755 s	0.1401 s	0.9228 s

tion based on two line-of-sight measurements in three dimensions as opposed to a search to obtain the (iterative) MLE of the Cartesian position. This also means the Jacobian of the converted Cartesian coordinates with respect to the original angle-only measurements can be calculated. In the case of ML, it is practically impossible to produce the Jacobian matrix as the "location" of the ML estimate's convergence point is not analytically related to the angle-only measurements. Furthermore, the proposed method is significantly faster than the ML estimate as no matrix multiplication is required, and the conversion is done in one step rather than requiring multiple iterations. In this section, the performance of the proposed method is compared to the ML method to verify the improvement in computation speed and examine the difference in standard deviation. Two experiments are made for comparison: one with the proposed method compared to one iteration of ILS in the ML method, and the other where the proposed method is compared to ten iterations of ILS in the ML method. The single iteration is the fastest the ML method can perform, but may lose some accuracy compared to using ten iterations to converge. Additionally, the methods are compared with more or fewer measurements. The same parameters are used from the previous simulations. Computation time is evaluated using MATLAB and is averaged over 100 Monte Carlo runs. The results are displayed in Table II and in Figs. 6 and 7. These figures present the time history of converted measurement errors along the trajectory (no filtering is carried out here).

The table shows that the computation time is significantly reduced by using the proposed method. The explicit conversion takes half the time compared to the ML method for one iteration, and hence naturally nearly 20 times less time for ten iterations. Additionally, the explicit conversion is nearly identical in performance to the ML method in terms of standard deviation and residual bias. The ML method is slightly better for situations with poor azimuth observability and situations with very good observability. The graph for the single-iteration ML method is neglected as it is nearly identical in performance to the ten-iteration ML method. The results show that the ML method can be favored in situations where computation speed is not a factor and an explicit expression of the conversion is not needed, but the explicit conversion is a very small reduction in performance if needed. Both methods have negligible bias, and

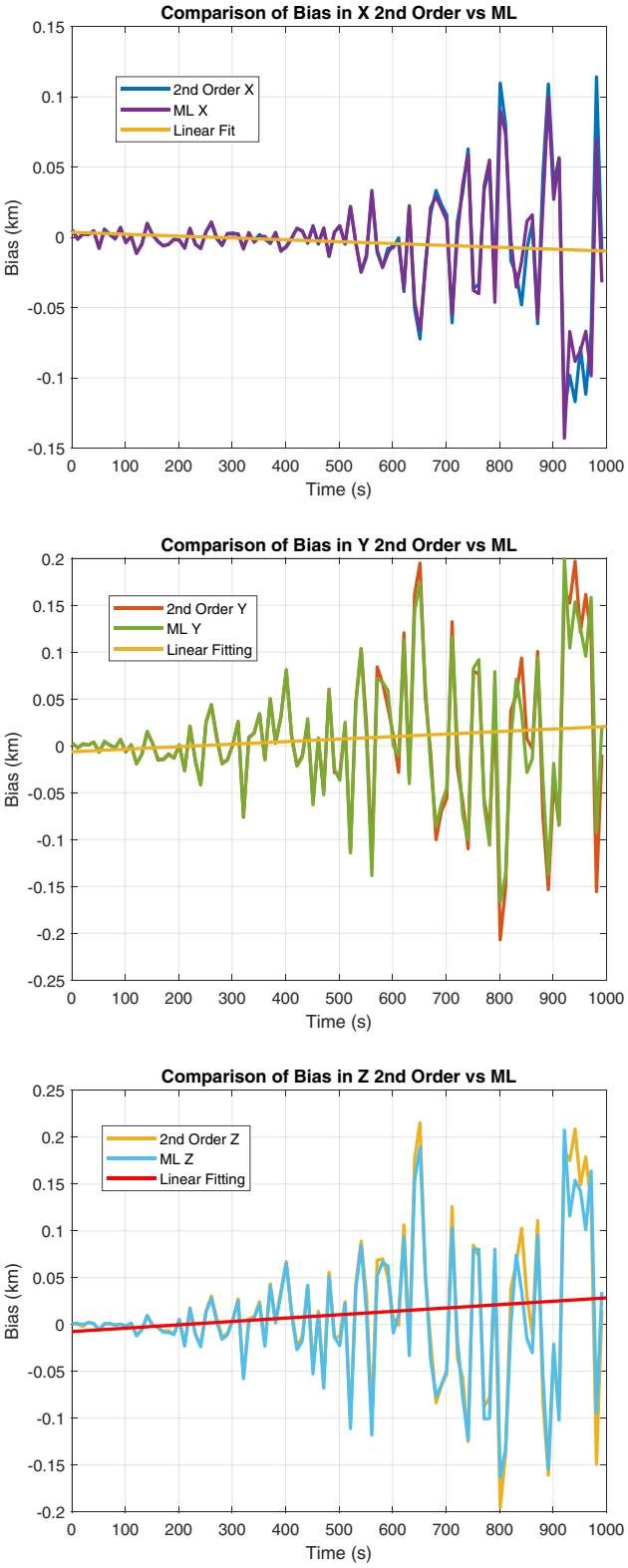


Fig. 6. Comparison of the biases in the proposed method and the ML method.

the residual bias is nearly the same. The results also imply that tracking methods, such as the EKF, can use these converted measurements without significant degradation compared to a mixed measurement filter tracking in Cartesian with angle-only measurments. Previous research in [11] has shown that in cases of high conversion

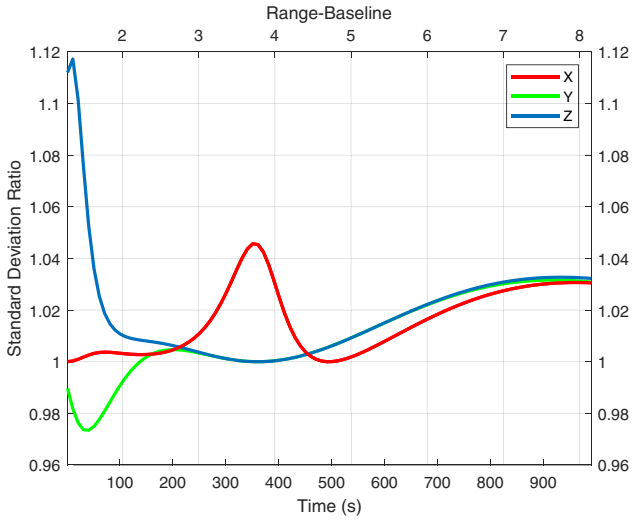


Fig. 7. Ratio of Cartesian coordinate standard deviations from the proposed method over the ML method using ten iterations.

bias significance (if done in the conventional way), the filter with measurements using the unbiased conversion performs better than the mixed measurement filter, as errors from nonlinearity can be present in the Jacobians.

V. CONCLUSION

When the observability of a target drops for passive sensors, it is necessary to account for higher order dynamics present in the conversion from angle-only measurements into Cartesian measurements. The bias present in the conversion can interfere with passive sensing applications such as target tracking and bias estimation. By using a second-order Taylor series expansion, it is possible to effectively remove the bias from converted measurements. Additionally, a more accurate model of the covariance of the converted noise is achieved. This method is useful as it is nearly equal in accuracy to ML methods, but also it is significantly faster and includes an explicit (noniterative) expression of the Cartesian position that can be used for applications. The bias present is relatively minimal and can be easily removed as well, meaning it is a robust conversion.

Future work with converted angle-only measurements would be to test this approach in situations incorporating additional real-world considerations. These would include asynchronous measurements and data association. Data association, in particular, is important as the method presented relies on correct associations for the formation of converted measurements. Asynchronous measurements must be propagated at the same time; thus, an additional source of error will be introduced. Therefore, the next step would be to integrate this approach to a target association method and determine how much error is likely from association errors, and how much error is obtained from the propagation of asynchronous measurements. Future work will also include analysis of the converted measurements with respect to an EKF, as this approach focuses on the conversion of individual measurements. Although the converted measurements have been shown to be effective relative to the ML solution, the inclusion of all of the nuances of tracking in an EKF such as process noise and target evolution models must be analyzed with respect to the coordinate conversion.

APPENDIX

A. *Introduction of the Conversion*

In order to calculate the unbiased conversion, it is necessary to derive the expressions for the converted measurements and then the derivatives of the converted measurements with respect to the angle measurements. First, the basics of the measurements are presented

$$x_s^t(k) = x^t(k) - x_s(k), \tag{32}$$

$$y_s^t(k) = y^t(k) - y_s(k), \tag{33}$$

$$z_s^t(k) = z^t(k) - z_s(k), \tag{34}$$

$$\alpha_s(k) = \text{atan2}\left(\frac{y_s^t(k)}{x_s^t(k)}\right), \tag{35}$$

$$\epsilon_s(k) = \text{atan2}\left(\frac{z_s^t(k)}{\sqrt{y_s^{t2}(k) + x_s^{t2}(k)}}\right). \tag{36}$$

The closest point of approach method involves converting the LOS measurements into a Cartesian ray. The ray for a sensor s is defined as

$$\mathbf{L}_s = \begin{bmatrix} \cos(\alpha_s)\cos(\epsilon_s) \\ \sin(\alpha_s)\cos(\epsilon_s) \\ \sin(\epsilon_s) \end{bmatrix}. \tag{37}$$

The sensor positions are assumed to be known, and the line between them is used to determine the closest point of approach

$$\mathbf{B}_{1,2} = \mathbf{x}_2 - \mathbf{x}_1 = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix}. \tag{38}$$

The closest point of approach for each sensor using the derivation in [1] and multiplied by the LOS ray to find the Cartesian positions on each LOS that are closest to each other. These are shifted by the sensor positions to place them within the same reference frame.

$$\mathbf{x}_1^{t,c}(k) = \begin{bmatrix} x_1^{t,c}(k) \\ y_1^{t,c}(k) \\ z_1^{t,c}(k) \end{bmatrix} = \mathbf{x}_1(k) + \frac{\mathbf{L}_1(k)(\mathbf{L}_1(k)'\mathbf{B}_{1,2}(k)) - (\mathbf{L}_1(k)'\mathbf{L}_2(k))(\mathbf{L}_2(k)'\mathbf{B}_{1,2}(k))}{1-(\mathbf{L}_1(k)'\mathbf{L}_2(k))^2}, \tag{39}$$

$$\mathbf{x}_2^{t,c}(k) = \begin{bmatrix} x_2^{t,c}(k) \\ y_2^{t,c}(k) \\ z_2^{t,c}(k) \end{bmatrix}$$

$$= \mathbf{x}_2(k) + \frac{\mathbf{L}_2(k)(\mathbf{L}_1(k)'\mathbf{L}_2(k))(\mathbf{L}_1(k)'\mathbf{B}_{1,2}(k)) - (\mathbf{L}_2(k)'\mathbf{B}_{1,2}(k))}{1-(\mathbf{L}_1(k)'\mathbf{L}_2(k))^2}, \quad (40)$$

$$\mathbf{x}_{1,2}^{t,c}(k) = \tfrac{1}{2}\left(\mathbf{x}_1^{t,c}(k) + \mathbf{x}_2^{t,c}(k)\right). \quad (41)$$

The conversion relies on the Cartesian vectors of the lines of sight defined as **L** and the line between the sensors defined as **B**. These are calculated as

$$\mathbf{L}_1(k) = \begin{bmatrix} \cos(\alpha_1(k))\cos(\epsilon_1(k)) \\ \sin(\alpha_1(k))\cos(\epsilon_1(k)) \\ \sin(\epsilon_1(k)) \end{bmatrix}, \quad (42)$$

$$\mathbf{L}_2(k) = \begin{bmatrix} \cos(\alpha_2(k))\cos(\epsilon_2(k)) \\ \sin(\alpha_2(k))\cos(\epsilon_2(k)) \\ \sin(\epsilon_2(k)) \end{bmatrix}, \quad (43)$$

$$\mathbf{B}_{1,2}(k) = \mathbf{x}_2(k) - \mathbf{x}_1(k) = \begin{bmatrix} x_2(k) - x_1(k) \\ y_2(k) - y_1(k) \\ z_2(k) - z_1(k) \end{bmatrix}. \quad (44)$$

For simplicity in the appendix the following substitutions are made:

$$N_1 = \mathbf{L}_1'\mathbf{B}_{1,2} - (\mathbf{L}_1'\mathbf{L}_2)(\mathbf{L}_2'\mathbf{B}_{1,2}), \quad (45)$$

$$N_2 = (\mathbf{L}_1'\mathbf{L}_2)(\mathbf{L}_1'\mathbf{B}_{1,2}) - \mathbf{L}_2'\mathbf{B}_{1,2}, \quad (46)$$

$$N_{x,1} = L_{x,1}N_1, \quad (47)$$

$$N_{y,1} = L_{y,1}N_1, \quad (48)$$

$$N_{z,1} = L_{z,1}N_1, \quad (49)$$

$$N_{x,2} = L_{x,2}N_2, \quad (50)$$

$$N_{y,2} = L_{y,2}N_2, \quad (51)$$

$$N_{z,2} = L_{z,2}N_2, \quad (52)$$

$$D = 1 - (\mathbf{L}_1'\mathbf{L}_2)^2, \quad (53)$$

which results in the equations

$$x_1^{t,c} = x_1 + \frac{N_{x,1}}{D}, \quad (54)$$

$$x_2^{t,c} = x_2 + \frac{N_{x,2}}{D}, \quad (55)$$

$$y_1^{t,c} = y_1 + \frac{N_{y,1}}{D}, \quad (56)$$

$$y_2^{t,c} = y_2 + \frac{N_{y,2}}{D}, \quad (57)$$

$$z_1^{t,c} = z_1 + \frac{N_{z,1}}{D}, \quad (58)$$

$$z_2^{t,c} = z_2 + \frac{N_{z,2}}{D}, \quad (59)$$

$$x_{1,2}^{t,c} = \frac{(x_1^{t,c} + x_2^{t,c})}{2}, \quad (60)$$

$$y_{1,2}^{t,c} = \frac{(y_1^{t,c} + y_2^{t,c})}{2}, \quad (61)$$

$$z_{1,2}^{t,c} = \frac{(z_1^{t,c} + z_2^{t,c})}{2}. \quad (62)$$

By making these substitutions, it is possible to use calculus rules to more efficiently represent the derivatives of the composite measurements.

B. Debiasing Calculation

It is necessary to calculate the bias for each Cartesian coordinate by using a Taylor series expansion to include the noise variables. For terseness, the symbol ζ is used to represent any of the Cartesian coordinates.

$$\zeta = x,\ y,\ z, \quad (63)$$

$$\begin{aligned} \zeta_{1,2}^{t,c,m}(k) \approx{}& \zeta^t(k) + \frac{\partial \zeta_{1,2}^{t,c}}{\partial \alpha_1} w_1^\alpha(k) + \frac{\partial \zeta_{1,2}^{t,c}}{\partial \alpha_2} w_2^\alpha(k) \\ &+ \frac{\partial \zeta_{1,2}^{t,c}}{\partial \epsilon_1} w_1^\epsilon(k) + \frac{\partial \zeta_{1,2}^{t,c}}{\partial \epsilon_2} w_2^\epsilon(k) \\ &+ 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1^2} w_1^\alpha(k)^2 + 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2^2} w_2^\alpha(k)^2 \\ &+ 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_1^2} w_1^\epsilon(k)^2 + 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_2^2} w_2^\epsilon(k)^2 \\ &+ \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \alpha_2} w_1^\alpha(k) w_2^\alpha(k) + \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_1} w_1^\alpha(k) w_1^\epsilon(k) \\ &+ \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_2} w_1^\alpha(k) w_2^\epsilon(k) + \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_1} w_2^\alpha(k) w_1^\epsilon(k) \\ &+ \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_2} w_2^\alpha(k) w_2^\epsilon(k) + \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_1 \partial \epsilon_2} w_1^\epsilon(k) w_2^\epsilon(k). \end{aligned} \quad (64)$$

The bias is defined as the difference between the truth and the expected value of the converted measurement. The first-order terms are eliminated from the bias, but the second-order terms contribute to the mean:

$$\begin{aligned} E[\zeta_{1,2}^{t,c,m}(k)] \approx{}& \zeta^t(k) + 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1^2}(\sigma_1^\alpha)^2 \\ &+ 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2^2}(\sigma_2^\alpha)^2 + 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_1^2}(\sigma_1^\epsilon)^2 \\ &+ 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_2^2}(\sigma_2^\epsilon)^2, \end{aligned} \quad (65)$$

$$c_{\zeta,1,2} = 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1^2}(\sigma_1^\alpha)^2 + 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2^2}(\sigma_2^\alpha)^2 + 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_1^2}(\sigma_1^\epsilon)^2 + 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_2^2}(\sigma_2^\epsilon)^2, \quad (66)$$

$$E[\zeta_{1,2}^{t,c,m}(k)] \approx \zeta^t(k) + c_{\zeta,1,2}, \quad (67)$$

$$\zeta_{1,2}^{t,c,m,db}(k) = \zeta_{1,2}^{t,c,m}(k) - c_{\zeta,1,2}, \quad (68)$$

$$E[\zeta_{1,2}^{t,c,m,db}(k)] \approx \zeta^t(k). \quad (69)$$

C. Variance Calculation

The variance is defined as the expected value of the measurement squared minus the mean squared:

$$\mathrm{Var}(\zeta_{1,2}^{t,c,m,db}(k)) = E[\zeta_{1,2}^{t,c,m,db}(k)^2] - E[\zeta_{1,2}^{t,c,m,db}(k)]^2. \quad (70)$$

The debiasing is included in order to find the variance of the debiased measurements

$$\begin{aligned}
&\mathrm{Var}(\zeta_{1,2}^{t,c,m,db}(k)) \\
&= E\Bigg[\Bigg(\zeta^t(k) + \frac{\partial \zeta_{1,2}^{t,c}}{\partial \alpha_1} w_1^\alpha(k) + \frac{\partial \zeta_{1,2}^{t,c}}{\partial \alpha_2} w_2^\alpha(k) \\
&\quad + \frac{\partial \zeta_{1,2}^{t,c}}{\partial \epsilon_1} w_1^\epsilon(k) + \frac{\partial \zeta_{1,2}^{t,c}}{\partial \epsilon_2} w_2^\epsilon(k) + 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1^2} w_1^\alpha(k)^2 \\
&\quad + 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2^2} w_2^\alpha(k)^2 + 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_1^2} w_1^\epsilon(k)^2 \\
&\quad + 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_2^2} w_2^\epsilon(k)^2 + \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \alpha_2} w_1^\alpha(k) w_2^\alpha(k) \\
&\quad + \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_1} w_1^\alpha(k) w_1^\epsilon(k) + \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_2} w_1^\alpha(k) w_2^\epsilon(k) \\
&\quad + \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_1} w_2^\alpha(k) w_1^\epsilon(k) + \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_2} w_2^\alpha(k) w_2^\epsilon(k) \\
&\quad + \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_1 \partial \epsilon_2} w_1^\epsilon(k) w_2^\epsilon(k) - c_{\zeta,1,2}\Bigg)^2\Bigg] - \zeta^t(k)^2.
\end{aligned} \quad (71)$$

The debiasing will remove some of the terms, so they are separated from the equation:

$$\begin{aligned}
&V(\zeta_{1,2}^{t,c,m,db}(k)) = \\
&\Bigg(\zeta^t(k) + 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1^2}(\sigma_1^\alpha)^2 + 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2^2}(\sigma_2^\alpha)^2 \\
&+0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_1^2}(\sigma_1^\epsilon)^2 + 0.5\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_2^2}(\sigma_2^\epsilon)^2 - c_{\zeta,1,2}\Bigg)^2 \\
&-\zeta^t(k)^2 + \left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \alpha_1}\right)^2(\sigma_1^\alpha)^2 + \left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \alpha_2}\right)^2(\sigma_2^\alpha)^2 \\
&+\left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \epsilon_1}\right)^2(\sigma_1^\epsilon)^2 + \left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \epsilon_2}\right)^2(\sigma_2^\epsilon)^2 \\
&+\left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \alpha_2}\right)^2(\sigma_1^\alpha)^2(\sigma_2^\alpha)^2 + \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_1}\right)^2(\sigma_1^\alpha)^2(\sigma_1^\epsilon)^2 \\
&+\left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_2}\right)^2(\sigma_1^\alpha)^2(\sigma_2^\epsilon)^2 + \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_1}\right)^2(\sigma_2^\alpha)^2(\sigma_1^\epsilon)^2 \\
&+\left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_2}\right)^2(\sigma_2^\alpha)^2(\sigma_2^\epsilon)^2 + \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_1 \partial \epsilon_2}\right)^2(\sigma_1^\epsilon)^2(\sigma_2^\epsilon)^2.
\end{aligned} \quad (72)$$

The final variance equation is as follows:

$$\begin{aligned}
&V(\zeta_{1,2}^{t,c,m,db}(k)) = \left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \alpha_1}\right)^2(\sigma_1^\alpha)^2 + \left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \alpha_2}\right)^2(\sigma_2^\alpha)^2 \\
&+\left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \epsilon_1}\right)^2(\sigma_1^\epsilon)^2 + \left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \epsilon_2}\right)^2(\sigma_2^\epsilon)^2 \\
&+\left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \alpha_2}\right)^2(\sigma_1^\alpha)^2(\sigma_2^\alpha)^2 + \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_1}\right)^2(\sigma_1^\alpha)^2(\sigma_1^\epsilon)^2 \\
&+\left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_2}\right)^2(\sigma_1^\alpha)^2(\sigma_2^\epsilon)^2 + \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_1}\right)^2(\sigma_2^\alpha)^2(\sigma_1^\epsilon)^2 \\
&+\left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_2}\right)^2(\sigma_2^\alpha)^2(\sigma_2^\epsilon)^2 + \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_1 \partial \epsilon_2}\right)^2(\sigma_1^\epsilon)^2(\sigma_2^\epsilon)^2.
\end{aligned} \quad (73)$$

D. Covariance Calculation

Similarly, the covariance is calculated for each combination of two Cartesian coordinates. As before, ζ is used to represent any particular Cartesian coordinate. γ is used to represent a different Cartesian coordinate. The calculation begins with the definition

$$\zeta = x,\ y,\ z, \quad (74)$$

$$\gamma = x,\ y,\ z, \quad (75)$$

$$\gamma \neq \zeta, \quad (76)$$

$$\begin{aligned}
&CV(\zeta_{1,2}^{t,c,m,db}(k), \gamma_{1,2}^{t,c,m,db}(k)) \\
&\quad = E[(\zeta_{1,2}^{t,c,m,db}(k) - E[\zeta_{1,2}^{t,c,m,db}(k)]) \\
&\quad\quad \times (\gamma_{1,2}^{t,c,m,db}(k) - E[\gamma_{1,2}^{t,c,m,db}(k)])] \\
&= E[\zeta_{1,2}^{t,c,m,db}(k)\gamma_{1,2}^{t,c,m,db}(k)] \\
&\quad\quad - E[\zeta_{1,2}^{t,c,m,db}(k)]E[\gamma_{1,2}^{t,c,m,db}(k)].
\end{aligned} \quad (77)$$

This is expanded with the second-order conversion

$$
\begin{aligned}
&\mathrm{Cov}(\zeta_{1,2}^{t,c,m,db}(k),\, \gamma_{1,2}^{t,c,m,db}(k)) \\
&= E\Bigg[\Bigg(\zeta^t(k) + \frac{\partial \zeta_{1,2}^{t,c}}{\partial \alpha_1} w_1^{\alpha}(k) + \frac{\partial \zeta_{1,2}^{t,c}}{\partial \alpha_2} w_2^{\alpha}(k) + \frac{\partial \zeta_{1,2}^{t,c}}{\partial \epsilon_1} w_1^{\epsilon}(k) \\
&\quad + \frac{\partial \zeta_{1,2}^{t,c}}{\partial \epsilon_2} w_2^{\epsilon}(k) + 0.5 \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1^2} w_1^{\alpha}(k)^2 + 0.5 \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2^2} w_2^{\alpha}(k)^2 \\
&\quad + 0.5 \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_1^2} w_1^{\epsilon}(k)^2 + 0.5 \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_2^2} w_2^{\epsilon}(k)^2 \\
&\quad + \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \alpha_2} w_1^{\alpha}(k) w_2^{\alpha}(k) + \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_1} w_1^{\alpha}(k) w_1^{\epsilon}(k) \\
&\quad + \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_2} w_1^{\alpha}(k) w_2^{\epsilon}(k) + \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_1} w_2^{\alpha}(k) w_1^{\epsilon}(k) \\
&\quad + \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_2} w_2^{\alpha}(k) w_2^{\epsilon}(k) + \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_1 \partial \epsilon_2} w_1^{\epsilon}(k) w_2^{\epsilon}(k) - c_{\zeta,1,2}\Bigg) \\
&\quad \times \Bigg(\gamma^t(k) + \frac{\partial \gamma_{1,2}^{t,c}}{\partial \alpha_1} w_1^{\alpha}(k) + \frac{\partial \gamma_{1,2}^{t,c}}{\partial \alpha_2} w_2^{\alpha}(k) + \frac{\partial \gamma_{1,2}^{t,c}}{\partial \epsilon_1} w_1^{\epsilon}(k) \\
&\quad + \frac{\partial \gamma_{1,2}^{t,c}}{\partial \epsilon_2} w_2^{\epsilon}(k) + 0.5 \frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_1^2} w_1^{\alpha}(k)^2 + 0.5 \frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_2^2} w_2^{\alpha}(k)^2 \\
&\quad + 0.5 \frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \epsilon_1^2} w_1^{\epsilon}(k)^2 + 0.5 \frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \epsilon_2^2} w_2^{\epsilon}(k)^2 \\
&\quad + \frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_1 \partial \alpha_2} w_1^{\alpha}(k) w_2^{\alpha}(k) + \frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_1} w_1^{\alpha}(k) w_1^{\epsilon}(k) \\
&\quad + \frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_2} w_1^{\alpha}(k) w_2^{\epsilon}(k) + \frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_1} w_2^{\alpha}(k) w_1^{\epsilon}(k) \\
&\quad + \frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_2} w_2^{\alpha}(k) w_2^{\epsilon}(k) + \frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \epsilon_1 \partial \epsilon_2} w_1^{\epsilon}(k) w_2^{\epsilon}(k) - c_{\gamma,1,2}\Bigg)\Bigg] \\
&\quad - \zeta^t(k)\gamma^t(k). \qquad (78)
\end{aligned}
$$

The debiasing will remove some of the terms, so they are separated from the equation:

$$
\begin{aligned}
&\mathrm{Cov}(\zeta_{1,2}^{t,c,m,db}(k),\, \gamma_{1,2}^{t,c,m,db}(k)) \\
&= \Bigg(\zeta^t(k) + 0.5 \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1^2} (\sigma_1^{\alpha})^2 + 0.5 \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2^2} (\sigma_2^{\alpha})^2 \\
&\quad + 0.5 \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_1^2} (\sigma_1^{\epsilon})^2 + 0.5 \frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_2^2} (\sigma_2^{\epsilon})^2 - c_{\zeta,1,2}\Bigg) \\
&\quad \times \Bigg(\gamma^t(k) + 0.5 \frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_1^2} (\sigma_1^{\alpha})^2 + 0.5 \frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_2^2} (\sigma_2^{\alpha})^2 \\
&\quad + 0.5 \frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \epsilon_1^2} (\sigma_1^{\epsilon})^2 + 0.5 \frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \epsilon_2^2} (\sigma_2^{\epsilon})^2 - c_{\gamma,1,2}\Bigg)
\end{aligned}
$$

$$
\begin{aligned}
&+ \left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \alpha_1}\right)\left(\frac{\partial \gamma_{1,2}^{t,c}}{\partial \alpha_1}\right)(\sigma_1^{\alpha})^2 + \left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \alpha_2}\right)\left(\frac{\partial \gamma_{1,2}^{t,c}}{\partial \alpha_2}\right)(\sigma_2^{\alpha})^2 \\
&+ \left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \epsilon_1}\right)\left(\frac{\partial \gamma_{1,2}^{t,c}}{\partial \epsilon_1}\right)(\sigma_1^{\epsilon})^2 + \left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \epsilon_2}\right)\left(\frac{\partial \gamma_{1,2}^{t,c}}{\partial \epsilon_2}\right)(\sigma_2^{\epsilon})^2 \\
&+ \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \alpha_2}\right)\left(\frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_1 \partial \alpha_2}\right)(\sigma_1^{\alpha})^2(\sigma_2^{\alpha})^2 \\
&+ \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_1}\right)\left(\frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_1}\right)(\sigma_1^{\alpha})^2(\sigma_1^{\epsilon})^2 \\
&+ \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_2}\right)\left(\frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_2}\right)(\sigma_1^{\alpha})^2(\sigma_2^{\epsilon})^2 \\
&+ \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_1}\right)\left(\frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_1}\right)(\sigma_2^{\alpha})^2(\sigma_1^{\epsilon})^2 \\
&+ \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_2}\right)\left(\frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_2}\right)(\sigma_2^{\alpha})^2(\sigma_2^{\epsilon})^2 \\
&+ \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_1 \partial \epsilon_2}\right)\left(\frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \epsilon_1 \partial \epsilon_2}\right)(\sigma_1^{\epsilon})^2(\sigma_2^{\epsilon})^2. \qquad (79)
\end{aligned}
$$

The final covariance equation is as follows:

$$
\begin{aligned}
&\mathrm{Cov}(\zeta_{1,2}^{t,c,m,db}(k),\, \gamma_{1,2}^{t,c,m,db}(k)) \\
&= \left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \alpha_1}\right)\left(\frac{\partial \gamma_{1,2}^{t,c}}{\partial \alpha_1}\right)(\sigma_1^{\alpha})^2 + \left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \alpha_2}\right)\left(\frac{\partial \gamma_{1,2}^{t,c}}{\partial \alpha_2}\right)(\sigma_2^{\alpha})^2 \\
&+ \left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \epsilon_1}\right)\left(\frac{\partial \gamma_{1,2}^{t,c}}{\partial \epsilon_1}\right)(\sigma_1^{\epsilon})^2 + \left(\frac{\partial \zeta_{1,2}^{t,c}}{\partial \epsilon_2}\right)\left(\frac{\partial \gamma_{1,2}^{t,c}}{\partial \epsilon_2}\right)(\sigma_2^{\epsilon})^2 \\
&+ \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \alpha_2}\right)\left(\frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_1 \partial \alpha_2}\right)(\sigma_1^{\alpha})^2(\sigma_2^{\alpha})^2 \\
&+ \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_1}\right)\left(\frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_1}\right)(\sigma_1^{\alpha})^2(\sigma_1^{\epsilon})^2 \\
&+ \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_2}\right)\left(\frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_1 \partial \epsilon_2}\right)(\sigma_1^{\alpha})^2(\sigma_2^{\epsilon})^2 \\
&+ \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_1}\right)\left(\frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_1}\right)(\sigma_2^{\alpha})^2(\sigma_1^{\epsilon})^2 \\
&+ \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_2}\right)\left(\frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \alpha_2 \partial \epsilon_2}\right)(\sigma_2^{\alpha})^2(\sigma_2^{\epsilon})^2 \\
&+ \left(\frac{\partial^2 \zeta_{1,2}^{t,c}}{\partial \epsilon_1 \partial \epsilon_2}\right)\left(\frac{\partial^2 \gamma_{1,2}^{t,c}}{\partial \epsilon_1 \partial \epsilon_2}\right)(\sigma_1^{\epsilon})^2(\sigma_2^{\epsilon})^2. \qquad (80)
\end{aligned}
$$

E. Derivatives of Converted Measurements

It is necessary to calculate the derivatives of the converted measurement with respect to the original measurements. This process begins with the first-order

derivatives. The derivatives are defined in terms of the substitutions made earlier. Furthermore, terms ϕ is included for terseness to represent any of the individual LOS measurements.

$$N_1 = \mathbf{L}_1'\mathbf{B}_{1,2} - (\mathbf{L}_1'\mathbf{L}_2)(\mathbf{L}_2'\mathbf{B}_{1,2}), \tag{81}$$

$$N_2 = (\mathbf{L}_1'\mathbf{L}_2)(\mathbf{L}_1'\mathbf{B}_{1,2}) - \mathbf{L}_2'\mathbf{B}_{1,2,} \tag{82}$$

$$\mathbf{N}_1 = N_1\mathbf{L}_1, \tag{83}$$

$$\mathbf{N}_2 = N_2\mathbf{L}_2, \tag{84}$$

$$D = 1 - (\mathbf{L}_1'\mathbf{L}_2)^2, \tag{85}$$

which results in the equations

$$\nabla\mathbf{N}_1 = \mathbf{L}_1'\nabla N_1 + N_1\nabla\mathbf{L}_1, \tag{86}$$

$$\nabla\mathbf{N}_2 = \mathbf{L}_2'\nabla N_2 + N_2\nabla\mathbf{L}_2, \tag{87}$$

$$\mathbf{x}_1^{t,c} = \mathbf{x}_1 + \frac{\mathbf{N}_1}{D}, \tag{88}$$

$$\mathbf{x}_2^{t,c} = \mathbf{x}_2 + \frac{\mathbf{N}_2}{D}, \tag{89}$$

$$\mathbf{x}_{1,2}^{t,c} = \frac{(\mathbf{x}_1^{t,c} + \mathbf{x}_2^{t,c})}{2}, \tag{90}$$

$$\nabla\mathbf{x}_{1,2}^{t,c} = \frac{1}{2}(\nabla\mathbf{x}_1^{t,c} + \nabla\mathbf{x}_2^{t,c}), \tag{91}$$

$$\nabla\mathbf{x}_1^{t,c} = \frac{D\nabla\mathbf{N}_1 - \mathbf{N}_1'\nabla D}{D^2}, \tag{92}$$

$$\nabla\mathbf{x}_2^{t,c} = \frac{D\nabla\mathbf{N}_2 - \mathbf{N}_2'\nabla D}{D^2}. \tag{93}$$

Gradients are used to simplify the equations, with an example being

$$\nabla\mathbf{x}_{1,2}^{t,c} = \begin{bmatrix} \frac{\partial x_{1,2}^{t,c}}{\partial\alpha_1} & \frac{\partial x_{1,2}^{t,c}}{\partial\epsilon_1} & \frac{\partial x_{1,2}^{t,c}}{\partial\alpha_2} & \frac{\partial x_{1,2}^{t,c}}{\partial\epsilon_2} \\ \frac{\partial y_{1,2}^{t,c}}{\partial\alpha_1} & \frac{\partial y_{1,2}^{t,c}}{\partial\epsilon_1} & \frac{\partial y_{1,2}^{t,c}}{\partial\alpha_2} & \frac{\partial y_{1,2}^{t,c}}{\partial\epsilon_2} \\ \frac{\partial z_{1,2}^{t,c}}{\partial\alpha_1} & \frac{\partial z_{1,2}^{t,c}}{\partial\epsilon_1} & \frac{\partial z_{1,2}^{t,c}}{\partial\alpha_2} & \frac{\partial z_{1,2}^{t,c}}{\partial\epsilon_2} \end{bmatrix}, \tag{94}$$

$$\nabla N_1 = \mathbf{B}_{1,2}'\nabla\mathbf{L}_1 - (\mathbf{L}_1'\mathbf{L}_2)(\mathbf{B}_{1,2}'\nabla\mathbf{L}_2) - (\mathbf{L}_2'\mathbf{B}_{1,2})(\mathbf{L}_1'\nabla\mathbf{L}_2 + \mathbf{L}_2'\nabla\mathbf{L}_1), \tag{95}$$

$$\nabla N_2 = -\mathbf{B}_{1,2}'\nabla\mathbf{L}_2 + (\mathbf{L}_1'\mathbf{L}_2)(\mathbf{B}_{1,2}'\nabla\mathbf{L}_1) + (\mathbf{L}_1'\mathbf{B}_{1,2})(\mathbf{L}_1'\nabla\mathbf{L}_2 + \mathbf{L}_2'\nabla\mathbf{L}_1), \tag{96}$$

$$\nabla D = -2(\mathbf{L}_1'\mathbf{L}_2)(\mathbf{L}_1'\nabla\mathbf{L}_2 + \mathbf{L}_2'\nabla\mathbf{L}_1), \tag{97}$$

$$\nabla\mathbf{L}_1 = \begin{bmatrix} \frac{\partial L_{x,1}}{\partial\alpha_1} & \frac{\partial L_{x,1}}{\partial\epsilon_1} & 0 & 0 \\ \frac{\partial L_{y,1}}{\partial\alpha_1} & \frac{\partial L_{y,1}}{\partial\epsilon_1} & 0 & 0 \\ \frac{\partial L_{z,1}}{\partial\alpha_1} & \frac{\partial L_{z,1}}{\partial\epsilon_1} & 0 & 0 \end{bmatrix}, \tag{98}$$

$$\nabla\mathbf{L}_2 = \begin{bmatrix} 0 & 0 & \frac{\partial L_{x,2}}{\partial\alpha_2} & \frac{\partial L_{x,2}}{\partial\epsilon_2} \\ 0 & 0 & \frac{\partial L_{y,2}}{\partial\alpha_2} & \frac{\partial L_{y,2}}{\partial\epsilon_2} \\ 0 & 0 & \frac{\partial L_{z,2}}{\partial\alpha_2} & \frac{\partial L_{z,2}}{\partial\epsilon_2} \end{bmatrix}, \tag{99}$$

$$\frac{\partial L_{x,1}}{\partial\alpha_1} = -\cos(\epsilon_1)\sin(\alpha_1), \tag{100}$$

$$\frac{\partial L_{x,1}}{\partial\epsilon_1} = -\sin(\epsilon_1)\cos(\alpha_1), \tag{101}$$

$$\frac{\partial L_{y,1}}{\partial\alpha_1} = \cos(\epsilon_1)\cos(\alpha_1), \tag{102}$$

$$\frac{\partial L_{y,1}}{\partial\epsilon_1} = -\sin(\epsilon_1)\sin(\alpha_1), \tag{103}$$

$$\frac{\partial L_{z,1}}{\partial\alpha_1} = 0, \tag{104}$$

$$\frac{\partial L_{z,1}}{\partial\epsilon_1} = \cos(\epsilon_1), \tag{105}$$

$$\frac{\partial L_{x,2}}{\partial\alpha_2} = -\cos(\epsilon_2)\sin(\alpha_2), \tag{106}$$

$$\frac{\partial L_{x,2}}{\partial\epsilon_2} = -\sin(\epsilon_2)\cos(\alpha_2), \tag{107}$$

$$\frac{\partial L_{y,2}}{\partial\alpha_2} = \cos(\epsilon_2)\cos(\alpha_2), \tag{108}$$

$$\frac{\partial L_{y,2}}{\partial\epsilon_2} = -\sin(\epsilon_2)\sin(\alpha_2), \tag{109}$$

$$\frac{\partial L_{z,2}}{\partial\alpha_2} = 0, \tag{110}$$

$$\frac{\partial L_{z,2}}{\partial\epsilon_2} = \cos(\epsilon_2), \tag{111}$$

The second-order derivatives are calculated to be

$$\phi = \alpha_1,\ \alpha_2,\ \epsilon_1,\ \epsilon_2, \tag{112}$$

$$\frac{\partial\nabla\mathbf{x}_{1,2}^{t,c}}{\partial\phi} = \frac{1}{2}(\frac{\partial\nabla\mathbf{x}_1^{t,c}}{\partial\phi} + \frac{\partial\nabla\mathbf{x}_2^{t,c}}{\partial\phi}), \tag{113}$$

$$\frac{\partial\nabla\mathbf{x}_1^{t,c}}{\partial\phi} = \left(\frac{D^2\left(\frac{\partial D}{\partial\phi}\nabla\mathbf{N}_1 + D\frac{\partial\nabla\mathbf{N}_1}{\partial\phi} - \mathbf{N}_1'\frac{\partial\nabla D}{\partial\phi} - \frac{\partial\mathbf{N}_1}{\partial\phi}'\nabla D\right)}{D^4}\right) + \left(\frac{-2D\frac{\partial D}{\partial\phi}(D\nabla\mathbf{N}_1 - \mathbf{N}_1'\nabla D)}{D^4}\right), \tag{114}$$

$$\frac{\partial\nabla\mathbf{x}_2^{t,c}}{\partial\phi} = \left(\frac{D^2\left(\frac{\partial D}{\partial\phi}\nabla\mathbf{N}_2 + D\frac{\partial\nabla\mathbf{N}_2}{\partial\phi} - \mathbf{N}_2'\frac{\partial\nabla D}{\partial\phi} - \frac{\partial\mathbf{N}_2}{\partial\phi}'\nabla D\right)}{D^4}\right) + \left(\frac{-2D\frac{\partial D}{\partial\phi}(D\nabla\mathbf{N}_2 - \mathbf{N}_2'\nabla D)}{D^4}\right), \tag{115}$$

$$\frac{\partial \nabla \mathbf{N}_1}{\partial \phi} = \mathbf{B}'_{1,2}\frac{\partial \nabla \mathbf{L}_1}{\partial \phi} - \left(\frac{\partial \mathbf{L}_1}{\partial \phi}'\mathbf{L}_2 + \mathbf{L}'_1\frac{\partial \mathbf{L}_2}{\partial \phi}\right)(\mathbf{B}'_{1,2}\nabla \mathbf{L}_2)$$
$$-(\mathbf{L}'_1\mathbf{L}_2)\left(\mathbf{B}'_{1,2}\frac{\partial \nabla \mathbf{L}_2}{\partial \phi}\right)$$
$$-\left(\frac{\partial \mathbf{L}_2}{\partial \phi}'\mathbf{B}_{1,2}\right)(\mathbf{L}'_1\nabla \mathbf{L}_2 + \mathbf{L}'_2\nabla \mathbf{L}_1)$$
$$-(\mathbf{L}'_2\mathbf{B}_{1,2})\left(\frac{\partial \mathbf{L}_1}{\partial \phi}'\nabla \mathbf{L}_2 + \frac{\partial \mathbf{L}_2}{\partial \phi}'\nabla \mathbf{L}_1\right.$$
$$\left.+\mathbf{L}'_1\frac{\partial \nabla \mathbf{L}_2}{\partial \phi} + \mathbf{L}'_2\frac{\partial \nabla \mathbf{L}_1}{\partial \phi}\right), \tag{116}$$

$$\frac{\partial \nabla \mathbf{N}_2}{\partial \phi} = -\mathbf{B}'_{1,2}\frac{\partial \nabla \mathbf{L}_2}{\partial \phi} + \left(\frac{\partial \mathbf{L}_1}{\partial \phi}'\mathbf{L}_2 + \mathbf{L}'_1\frac{\partial \mathbf{L}_2}{\partial \phi}\right)(\mathbf{B}'_{1,2}\nabla \mathbf{L}_1)$$
$$+(\mathbf{L}'_1\mathbf{L}_2)\left(\mathbf{B}'_{1,2}\frac{\partial \nabla \mathbf{L}_1}{\partial \phi}\right)$$
$$+\left(\frac{\partial \mathbf{L}_1}{\partial \phi}'\mathbf{B}_{1,2}\right)(\mathbf{L}'_1\nabla \mathbf{L}_2 + \mathbf{L}'_2\nabla \mathbf{L}_1)$$
$$+(\mathbf{L}'_1\mathbf{B}_{1,2})\left(\frac{\partial \mathbf{L}_1}{\partial \phi}'\nabla \mathbf{L}_2 + \frac{\partial \mathbf{L}_2}{\partial \phi}'\nabla \mathbf{L}_1\right.$$
$$\left.+\mathbf{L}'_1\frac{\partial \nabla \mathbf{L}_2}{\partial \phi} + \mathbf{L}'_2\frac{\partial \nabla \mathbf{L}_1}{\partial \phi}\right), \tag{117}$$

$$\frac{\partial \nabla D}{\partial \phi} = -2\left(\frac{\partial \mathbf{L}_1}{\partial \phi}'\mathbf{L}_2 + \mathbf{L}'_1\frac{\partial \mathbf{L}_2}{\partial \phi}\right)(\mathbf{L}'_1\nabla \mathbf{L}_2 + \mathbf{L}'_2\nabla \mathbf{L}_1)$$
$$-2(\mathbf{L}'_1\mathbf{L}_2)\left(\frac{\partial \mathbf{L}_1}{\partial \phi}'\nabla \mathbf{L}_2 + \frac{\partial \mathbf{L}_2}{\partial \phi}'\nabla \mathbf{L}_1\right.$$
$$\left.+\mathbf{L}'_1\frac{\partial \nabla \mathbf{L}_2}{\partial \phi} + \mathbf{L}'_2\frac{\partial \nabla \mathbf{L}_1}{\partial \phi}\right), \tag{118}$$

$$\frac{\partial \nabla \mathbf{L}_1}{\partial \phi} = \begin{bmatrix} \frac{\partial^2 L_{x,1}}{\partial \alpha_1 \partial \phi} & \frac{\partial^2 L_{x,1}}{\partial \epsilon_1 \partial \phi} & 0 & 0 \\ \frac{\partial^2 L_{y,1}}{\partial \alpha_1 \partial \phi} & \frac{\partial^2 L_{y,1}}{\partial \epsilon_1 \partial \phi} & 0 & 0 \\ \frac{\partial^2 L_{z,1}}{\partial \alpha_1 \partial \phi} & \frac{\partial^2 L_{z,1}}{\partial \epsilon_1 \partial \phi} & 0 & 0 \end{bmatrix}, \tag{119}$$

$$\frac{\partial \nabla \mathbf{L}_2}{\partial \phi} = \begin{bmatrix} 0 & 0 & \frac{\partial^2 L_{x,2}}{\partial \alpha_2 \partial \phi} & \frac{\partial^2 L_{x,2}}{\partial \epsilon_2 \partial \phi} \\ 0 & 0 & \frac{\partial^2 L_{y,2}}{\partial \alpha_2 \partial \phi} & \frac{\partial^2 L_{y,2}}{\partial \epsilon_2 \partial \phi} \\ 0 & 0 & \frac{\partial^2 L_{z,2}}{\partial \alpha_2 \partial \phi} & \frac{\partial^2 L_{z,2}}{\partial \epsilon_2 \partial \phi} \end{bmatrix}, \tag{120}$$

$$\frac{\partial^2 L_{x,1}}{\partial^2 \alpha_1} = -\cos(\epsilon_1)\cos(\alpha_1), \tag{121}$$
$$\frac{\partial^2 L_{x,1}}{\partial^2 \epsilon_1} = -\cos(\epsilon_1)\cos(\alpha_1), \tag{122}$$
$$\frac{\partial^2 L_{y,1}}{\partial^2 \alpha_1} = -\cos(\epsilon_1)\sin(\alpha_1), \tag{123}$$
$$\frac{\partial^2 L_{y,1}}{\partial^2 \epsilon_1} = -\cos(\epsilon_1)\sin(\alpha_1), \tag{124}$$
$$\frac{\partial^2 L_{z,1}}{\partial^2 \alpha_1} = 0, \tag{125}$$
$$\frac{\partial^2 L_{z,1}}{\partial^2 \epsilon_1} = -\sin(\epsilon_1), \tag{126}$$
$$\frac{\partial^2 L_{x,1}}{\partial \alpha_1 \partial \epsilon_1} = \sin(\epsilon_1)\sin(\alpha_1), \tag{127}$$
$$\frac{\partial^2 L_{y,1}}{\partial \alpha_1 \partial \epsilon_1} = -\sin(\epsilon_1)\cos(\alpha_1), \tag{128}$$
$$\frac{\partial^2 L_{z,1}}{\partial \alpha_1 \partial \epsilon_1} = 0, \tag{129}$$
$$\frac{\partial^2 L_{x,1}}{\partial \alpha_1 \partial \alpha_2} = 0, \tag{130}$$
$$\frac{\partial^2 L_{x,1}}{\partial \alpha_1 \partial \epsilon_2} = 0, \tag{131}$$
$$\frac{\partial^2 L_{y,1}}{\partial \alpha_1 \partial \alpha_2} = 0, \tag{132}$$
$$\frac{\partial^2 L_{y,1}}{\partial \alpha_1 \partial \epsilon_2} = 0, \tag{133}$$
$$\frac{\partial^2 L_{z,1}}{\partial \alpha_1 \partial \alpha_2} = 0, \tag{134}$$
$$\frac{\partial^2 L_{z,1}}{\partial \alpha_1 \partial \epsilon_2} = 0, \tag{135}$$
$$\frac{\partial^2 L_{x,1}}{\partial \epsilon_1 \partial \alpha_2} = 0, \tag{136}$$
$$\frac{\partial^2 L_{x,1}}{\partial \epsilon_1 \partial \epsilon_2} = 0, \tag{137}$$
$$\frac{\partial^2 L_{y,1}}{\partial \epsilon_1 \partial \alpha_2} = 0, \tag{138}$$
$$\frac{\partial^2 L_{y,1}}{\partial \epsilon_1 \partial \epsilon_2} = 0, \tag{139}$$
$$\frac{\partial^2 L_{z,1}}{\partial \epsilon_1 \partial \alpha_2} = 0, \tag{140}$$
$$\frac{\partial^2 L_{z,1}}{\partial \epsilon_1 \partial \epsilon_2} = 0, \tag{141}$$
$$\frac{\partial^2 L_{x,2}}{\partial^2 \alpha_2} = -\cos(\epsilon_2)\cos(\alpha_2), \tag{142}$$
$$\frac{\partial^2 L_{x,2}}{\partial^2 \epsilon_2} = -\cos(\epsilon_2)\cos(\alpha_2), \tag{143}$$
$$\frac{\partial^2 L_{y,2}}{\partial^2 \alpha_2} = -\cos(\epsilon_2)\sin(\alpha_2), \tag{144}$$
$$\frac{\partial^2 L_{y,2}}{\partial^2 \epsilon_2} = -\cos(\epsilon_2)\sin(\alpha_2), \tag{145}$$
$$\frac{\partial^2 L_{z,2}}{\partial^2 \alpha_2} = 0, \tag{146}$$

$$\frac{\partial^2 L_{z,2}}{\partial^2 \epsilon_2} = -\sin(\epsilon_2), \tag{147}$$

$$\frac{\partial^2 L_{x,2}}{\partial \alpha_2 \partial \epsilon_2} = \sin(\epsilon_2)\sin(\alpha_2), \tag{148}$$

$$\frac{\partial^2 L_{y,2}}{\partial \alpha_2 \partial \epsilon_2} = -\sin(\epsilon_2)\cos(\alpha_2), \tag{149}$$

$$\frac{\partial^2 L_{z,2}}{\partial \alpha_2 \partial \epsilon_2} = 0, \tag{150}$$

$$\frac{\partial^2 L_{x,2}}{\partial \alpha_1 \partial \alpha_2} = 0, \tag{151}$$

$$\frac{\partial^2 L_{x,2}}{\partial \alpha_1 \partial \epsilon_2} = 0, \tag{152}$$

$$\frac{\partial^2 L_{y,2}}{\partial \alpha_1 \partial \alpha_2} = 0, \tag{153}$$

$$\frac{\partial^2 L_{y,2}}{\partial \alpha_1 \partial \epsilon_2} = 0, \tag{154}$$

$$\frac{\partial^2 L_{z,2}}{\partial \alpha_1 \partial \alpha_2} = 0, \tag{155}$$

$$\frac{\partial^2 L_{z,2}}{\partial \alpha_1 \partial \epsilon_2} = 0, \tag{156}$$

$$\frac{\partial^2 L_{x,2}}{\partial \epsilon_1 \partial \alpha_2} = 0, \tag{157}$$

$$\frac{\partial^2 L_{x,2}}{\partial \epsilon_1 \partial \epsilon_2} = 0, \tag{158}$$

$$\frac{\partial^2 L_{y,2}}{\partial \epsilon_1 \partial \alpha_2} = 0, \tag{159}$$

$$\frac{\partial^2 L_{y,2}}{\partial \epsilon_1 \partial \epsilon_2} = 0, \tag{160}$$

$$\frac{\partial^2 L_{z,2}}{\partial \epsilon_1 \partial \alpha_2} = 0, \tag{161}$$

$$\frac{\partial^2 L_{z,2}}{\partial \epsilon_1 \partial \epsilon_2} = 0. \tag{162}$$

REFERENCES

[1] "Closest point between two rays." Accessed: Jul. 25, 2019. [Online]. Available: http://morroworks.palitri.com/Content/Docs/Rays%20closest%20point.pdf.

[2] V. Aidala
"Kalman filter behavior in bearings-only tracking applications,"
IEEE Trans. Aerosp. Electron. Syst., vol. AES-15, no. 1, pp. 29–39, Jan. 1979.

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

[4] 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.

[5] H. Chen and F. Han
"Bias estimation for multiple passive sensors,"
Proc. Int. Conf. Measurement, Inf. Control, 2012, pp. 1081–1084.

[6] D. F. Crouse
"Cubature/unscented/sigma point Kalman filtering with angular measurement models,"
Proc. 18th Int. Conf. Inf. Fusion, 2015, pp. 1550–1557.

[7] B. Davis and W. D. Blair
"Gaussian mixture approach to long range radar tracking with high range resolution," in
Proc. IEEE Aerosp. Conf., 2015, pp. 1–9.

[8] Á. F. García-Fernández, S. Maskell, P. Horridge, and J. Ralph
"Gaussian tracking with Kent-distributed direction-of-arrival measurements,"
IEEE Trans. Veh. Technol., vol. 70, no. 7, pp. 7249–7254, Jul. 2021.

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

[10] 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,"
Proc. SPIE, vol. 11018, 2019, Art. no. 1101805.

[11] D. Lerro and Y. Bar-Shalom
"Tracking with debiased consistent converted measurements versus EKF,"
IEEE Trans. Aerosp. Electron. Syst., vol. 29, no. 3, pp. 1015–1022, Jul. 1993.

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

[13] N. Okello and B. Ristic
"Maximum likelihood registration for multiple dissimilar sensors,"
IEEE Trans. Aerosp. Electron. Syst., vol. 39, pp. 1074–1083, Aug. 2003.

[14] R. 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.

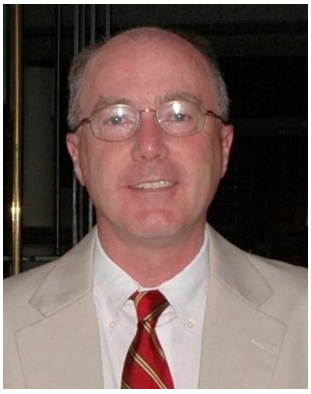
[15] Z. Sutton, P. Willett, T. Fair, and Y. Bar-Shalom
"MLPMH tracking in 3 dimensions using cluttered measurements from multiple 2-dimensional sensors,"
J. Adv. Inf. Fusion, vol. 16, no. 2, pp. 92–113, Dec. 2021.

[16] 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.

[17] X. Tian and Y. Bar-Shalom
"Coordinate conversion and tracking for very long range radars,"
IEEE Trans. Aerosp. Electron. Syst., vol. 45, no. 3, pp. 1073–1088, Jul. 2009.

Michael Kowalski received the B.Sc. and Ph.D. degrees in electrical engineering from the University of Connecticut, Storrs, CT, USA, in 2015 and 2020, respectively. He worked with his advisors Dr. Peter Willett and Dr. Yaakov Bar-Shalom in pursuit of his Ph.D. His research focuses on bias estimation as well as sensor fusion and target tracking. In August 2020, he joined the Georgia Tech Research Institute and is currently supporting the development of assessment tools and metrics in the Benchmark environment.

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



Peter Willett (F'03) had been a faculty member in the Electrical and Computer Engineering Department, University of Connecticut, Storrs, CT, USA, since 1986. In 1998, he became a Professor. Since 2003, he has been an IEEE Fellow. His primary areas of research have been statistical signal processing, detection, machine learning, communications, data fusion, and tracking. He was the Chief Editor for the IEEE Aerospace and Electronic Systems Magazine from 2018 to 2020. He was the Editor-in-Chief for the IEEE Transactions on Aerospace and Electronic Systems from 2006 to 2011 and for the IEEE Signal Processing Letters from 2014 to 2016. 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 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.

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