June 2024 Volume 19 Number 1 ISSN 1557-6418

JAIF24

Journal of Advances in Information Fusion

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

Regular Papers Page

ISIF

INTERNATIONAL SOCIETY OF INFORMATION FUSION

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

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

A Single-Pass Noise Covariance Estimation Algorithm in Nonswitching Multiple-Model Adaptive Kalman Filters for Nonstationary Systems

HEE-SEUNG KIM
ADAM BIENKOWSKI
LINGYI ZHANG
KRISHNA R. PATTIPATI

This paper presents a single-pass stochastic gradient descent (SGD) algorithm for estimating unknown noise covariances. The proposed algorithm is designed for nonswitching multiple-model adaptive Kalman filters, where the noise covariances can occasionally jump up or down by an unknown magnitude. Compared to our previous batch estimation or multipass decision-directed estimation methods, the proposed algorithm has the advantage of reading measurement data exactly once, leading to a significant improvement in computational efficiency and practicality. Moreover, the algorithm achieves an acceptable level of root mean square error (RMSE) in state estimates, making it suitable for real-time industrial applications. The proposed algorithm utilizes recursive fading memory estimates of the sample cross-correlations of the innovations and employs the root mean square propagation (RMSprop) accelerated SGD algorithm. The combination of these techniques enables the algorithm to achieve high accuracy in estimating the unknown noise covariances while maintaining superior computational efficiency over iterative batch methods. The paper presents a comparative evaluation of the proposed method on several test cases, which demonstrate its effectiveness in terms of computational efficiency and estimation accuracy. Overall, the proposed algorithm offers a promising approach for real-time noise covariance estimation in multiple-model adaptive Kalman filters.

Manuscript received February 5, 2024; revised May 21, 2024; released for publication October 14, 2024

Refereeing of this contribution was handled by Erik Leitinger.

This work was supported in part by the U.S. Office of Naval Research (ONR), in part by the U.S. Naval Research Laboratory (NRL) under Grant N00014-18-1-1238, N00014-21-1-2187 and Grant N00173-16-1-G905. This paper was presented in part at the Twenty-Fourth International Conference on Information Fusion, Sun City, South Africa, November, 2021.

The authors are with Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269, USA, (e-mail: hee-seung.kim@uconn.edu; adam.bienkowski@uconn.edu; lingyi.zhang@uconn.edu; krishna.pattipati@uconn.edu). (Corresponding author: Hee-Seung Kim.)

1557-6418/2024/$17.00

I. INTRODUCTION

The Kalman filter (KF) [13] is a widely used algorithm that provides an optimal minimum mean square error (MMSE) estimate for discrete-time linear dynamic systems under the Gaussian assumption, provided that the mean and covariance are known. When the noises entering a system are non-Gaussian but still have known first and second moments, i.e., mean and covariance, the KF remains the best linear state estimator due to its ability to efficiently incorporate prior knowledge of the system dynamics and noise statistics. As a consequence of numerous research studies, the KF has gained significant interest and attention in many industrial applications, including fault diagnosis, robotics, signal processing, navigation, and target tracking, to name a few [2], [3]. However, in many real-world circumstances, the statistics of noise processes are either completely unknown or partially known.

In order to estimate unknown noise covariance parameters, Zhang *et al*. [32] derived the necessary and sufficient conditions for their identifiability, and then proposed an iterative *batch* optimization algorithm that ensures uncorrelated innovations. The rank of a matrix formed from the cross-correlations of the weighted sum of innovations, where the weights are the coefficients of the minimal polynomial of any closed-loop filter matrix, optimal or suboptimal, is required for the noise covariance identifiability. The innovation sequence of an optimal KF under the Gaussian assumption is orthogonal, indicating that the innovations are strictly white and independent of each other [3]. Zhang *et al*. [32] formulated an objective function using normalized temporal cross-correlations of the innovations based on this attribute to determine the optimal gain and subsequently the innovation (preresidual) and postresidual covariances, the measurement noise covariance and the process noise covariance.

We presented an improved method for estimating noise covariances, employing a sequential mini-batch stochastic gradient descent (SGD) algorithm that requires multiple passes through the data. Moreover, we showed a technique to detect changes in noise covariances when applying this estimation method to nonstationary systems [17]. To avoid multiple passes through the data required by the batch and multipass algorithms, we propose herein a *single-pass* real-time adaptive Kalman filtering approach designed for nonstationary systems. The proposed method is suitable for

scenarios in which process and measurement noise covariances occasionally fluctuate by an unknown magnitude and the system behavior is associated with one of a finite number of known models.

A. Prior Work

The relationship between the covariance of the state estimation error and the innovations in any suboptimal filter is the key to process and measurement noise covariance estimation. This relationship serves as a fundamental building block for correlation-based approaches. Pioneering contributions using this approach were made by [5], [21], [24], [25].

In linear state space models, Sarkka and Nummenmaa [26] presented a variational Bayesian approach for the joint recursive estimate of the dynamic state and measurement noise parameters. The method is implemented by forming separable variance approximations to the joint posterior distribution of state and noise parameters at each time step. This approach, however, does not take variations in process noise into consideration. Because of their mode-seeking behavior, the variational algorithms often converge to local minima and typically need tuning parameters to converge to the correct parameters.

Our sequential mini-batch estimation method [17] enhanced the computational efficiency and accuracy of the batch estimation algorithm in [32] by applying dynamic convergence thresholds and adaptive step size rules. To update the filter gain, we used sequential fading memory mini-batch estimates of the innovation correlations. For nonstationary systems, a change-point detection algorithm described in [15] was used for determining the time points of abrupt changes in unknown noise covariances based on the innovation sequence.

In the multiple-model adaptive estimation method proposed herein, the system is assumed to obey one of a finite number of models, and each model has its own nonswitching dynamics [3]. The overall estimate of the system state is obtained by taking a convex combination of the estimates from multiple parallel filters. The weights used in the convex combination correspond to the posterior model probabilities.

The following are the limits of previous research [17] on noise covariance estimation in nonstationary systems. First, the previous methods are computationally expensive because they require multiple passes through the observation data and are not suitable for online streaming data applications. Second, since the sequential estimation method is used for samples between two consecutive change points, the accuracy of the decision-directed noise covariance estimation method is reliant on the accuracy of the change-point detection algorithm. Third, the previous methods assumed that the structure of the dynamic model was known. In this paper, we relax this assumption and propose a streaming algorithm that extends the method to multiple-model settings.

B. Contribution and Organization of the Paper

We present a single-pass sequential mini-batch noise covariance estimation algorithm suitable for streaming data as an extension of the work in [17], [18] for nonstationary and nonswitching multiple-model systems. Our proposed method enables the estimation of the measurement and process noise covariances without the use of a change-point detection algorithm. We enhance the computational efficiency of the method via a single-pass through the observation data. The only caveat is that jumps are assumed to occur occasionally, and after the filter has reached a steady state, that is, the jumps are infrequent. However, small variations in the noise covariances are allowed in between jumps. More significantly, the structure of the dynamic model is unknown, but is assumed to belong to one of a finite number of known models.

We validate the proposed method on several nonstationary and multiple-model system test cases. In addition, we derive noise covariance identifiability conditions in terms of prefit residual (innovation) correlations as in [32], as well as postfit residual correlations and output correlations; the latter was used in the covariance estimation algorithm in [25]. We also prove the convergence of the iterative algorithm for process noise covariance, which was not established in our prior work.

The paper is organized as follows: In Section 2, we provide an overview of the multiple-model KF. Then, in Section 3, we discuss the identifiability conditions in terms of prefit and postfit residuals and outputs for estimating the unknown noise covariances in each individual model. Section 4 provides approaches for obtaining the unknown covariance parameters in a multiple-model system using the sequential mini-batch SGD method, including the fading memory filter-based correlation estimation, and the SGD update of the Kalman gain. In Section 5, numerical results[1] show the evidence that our method can track unknown noise covariances in nonstationary systems, as well as systems exhibiting dynamics from a finite number of known models, and that the single-pass algorithm is computationally efficient. Lastly, we conclude the paper and discuss potential avenues for future work in Section 6.

II. PLANT AND MEASUREMENT MODEL FOR THE MULTIPLE-MODEL KF

The multiple-model approach assumes that the system obeys one of a finite number of fixed models. Formally, the approach assumes that the linear discrete-time stochastic dynamic system can assume one of J models, $j = 1, 2, ..., J$, given by

$$x(k+1) = F^j x(k) + \Gamma^j v^j(k), \tag{1}$$

[1]Numerical results suggest that the occasional jump assumption may be relaxed in practice.

$$z(k) = H^j x(k) + w^j(k), \tag{2}$$

where $x(k)$ is the n_x-dimensional state vector, $z(k)$ is the n_z-dimensional measurement vector, and j is the candidate model. Here F^j and H^j are the $n_x \times n_x$ state transition matrix and the $n_z \times n_x$ measurement matrix of the system, respectively, and Γ^j is the noise gain matrix. We assume that the process noise $v^j(k)$ and the measurement noise $w^j(k)$ processes are the sequences of zero-mean white Gaussian noises with *unknown* process noise covariance $Q^j(k)$ and *unknown* measurement noise covariance $R^j(k)$, respectively. Note that the initial state error and the two noise processes are assumed to be mutually independent. We assume that $Q^j(k)$ and $R^j(k)$ are piecewise constants such that the filter reaches a steady state between any two jumps and that the jump is of an unknown magnitude.

Given $Q^j(k)$ and $R^j(k)$, the multiple-model adaptive KF involves the consecutive processes of prediction and update given by [3], [12], [13], [20], [27]

$$\hat{x}^j(k+1|k) = F^j \hat{x}^j(k|k), \tag{3}$$

$$v^j(k+1) = z(k+1) - H^j \hat{x}^j(k+1|k), \tag{4}$$

$$\hat{x}^j(k+1|k+1) = \hat{x}^j(k+1|k) + W^j(k+1) v^j(k+1), \tag{5}$$

$$P^j(k+1|k) = F^j P^j(k|k) F^{j\prime} + \Gamma^j Q^j(k) \Gamma^{j\prime}, \tag{6}$$

$$S^j(k+1) = H^j P^j(k+1|k) H^{j\prime} + R^j(k), \tag{7}$$

$$W^j(k+1) = P^j(k+1|k) H^{j\prime} S^j(k+1)^{-1}, \tag{8}$$

$$\begin{aligned} P^j(k+1|k+1) &= (I_{n_x} - W^j(k+1)H^j) P^j(k+1|k) \\ &\quad (I_{n_x} - W^j(k+1)H^j)' + W^j(k+1) R^j(k) W^j(k+1)', \end{aligned} \tag{9}$$

$$\Lambda^j(k) = \frac{1}{\sqrt{|2\pi S^j(k)|}} \exp\left(-\frac{1}{2} v^j(k)' S^j(k)^{-1} v^j(k)\right), \tag{10}$$

$$p^j(k) = \frac{\Lambda^j(k) p^j(k-1)}{\sum_{l=1}^{J} \Lambda^l(k) p^l(k-1)}. \tag{11}$$

The KF predicts the next state estimate at time index $(k+1)$, given the observations up to time index k in (3) and the concomitant predicted state estimation error covariance in (6), using model-specific system dynamics, the updated state error covariance $P^j(k|k)$ at time index k and the process noise covariance, $Q^j(k)$. The updated state estimate at time $(k+1)$ in (5) incorporates the measurement at time $(k+1)$ via the Kalman gain matrix in (8), which depends on the innovation covariance $S^j(k+1)$ (which in turn depends on the measurement noise covariance $R^j(k)$ and the predicted state error covariance $P^j(k+1|k)$). The updated state error covariance $P^j(k+1|k+1)$ is computed via (9). This corresponds to Joseph form in [3], [8], which guarantees that the updated state covariance matrix will remain positive definite.

The mode likelihood function $\Lambda^j(k)$ is computed via (10), which depends on the innovation sequence $v^j(k)$ and innovation covariance $S^j(k)$. In (10), $|\cdot|$ is the determinant, and the determinant of any scalar value times a matrix is equal to the determinant of the matrix times the scalar raised to the dimension of the matrix. This means that $|2\pi S^j(k)| = (2\pi)^{n_z} |S^j(k)|$ for a multidimensional random variable. Note that n_z here is the dimension of the measurement (or innovation) vector. The mode probability $p^j(k)$ corresponding to each candidate model at time index k is computed via (11). Without loss of generality, we assume the initial mode probability $p^j(0) = 1/J$.

III. NECESSARY AND SUFFICIENT CONDITIONS FOR THE IDENTIFIABILITY OF UNKNOWN COVARIANCES

We derive the necessary and sufficient conditions to estimate the unknown covariance matrices in terms of prefit residual (innovation) correlations, as well as postfit residual correlations and output correlations. Note that the identifiability conditions of the multiple-model approach depend on each candidate model since the corresponding KFs are noninteracting.

A. Innovation-Based Identifiability Conditions

Consider model j. Assume that Q^j and R^j are unknown but are piecewise constants such that the filter reaches the steady state before any jump to a new value (in practice, they can vary in between jumps as demonstrated in illustrative examples). Let us define the coefficients of the m^{th} order minimal polynomial of the closed-loop filter matrix $\bar{F}^j$, $\sum_{i=0}^{m} a_i^j (\bar{F}^j)^{m-i} = 0$, $a_0^j = 1$. Now, consider the innovations corresponding to a stable, suboptimal closed-loop filter matrix $\bar{F}^j = F^j(I_{n_x} - W^j H^j)$ given by [29], [32]

$$\begin{aligned} v^j(k) &= H^j (\bar{F}^j)^m \tilde{x}^j(k-m|k-m-1) + H^j \sum_{\ell=0}^{m-1} \Big\{ (\bar{F}^j)^{m-1-\ell} \\ &\quad \times \left[\Gamma^j v^j(k-m+\ell) - F^j W^j w^j(k-m+\ell) \right] \Big\} + w^j(k), \end{aligned} \tag{12}$$

where $\tilde{x}^j(k-m|k-m-1) = x^j(k-m) - \hat{x}^j(k-m|k-m-1)$ is the predicted error at time $(k-m)$. Given the innovation sequence (12), a weighted sum of innovations,

$\xi^j(k)$, is obtained as follows:

$$\xi^j(k) = \sum_{i=0}^{m} a_i^j v^j(k-i)$$
$$= \sum_{l=1}^{m} B_l^j v^j(k-l) + \sum_{l=0}^{m} G_l^j w^j(k-l). \quad (13)$$

It is easy to see that $\xi^j(k)$ is the sum of two moving average processes driven by the process noise and measurement noise, respectively [29], [32]. Here, B_l^j and G_l^j are given by

$$B_l^j = H^j\Big(\sum_{i=0}^{l-1} a_i^j (\bar{F}^j)^{l-i-1}\Big)\Gamma^j, \quad (14)$$

$$G_l^j = \Big[a_l^j I_{n_z} - H^j\Big(\sum_{i=0}^{l-1} a_i^j (\bar{F}^j)^{l-i-1}\Big)F^j W^j\Big],\ G_0^j = I_{n_z}. \quad (15)$$

Then, if we define the cross-covariance between $\xi^j(k)$ and $\xi^j(k-\ell)$ as L_ℓ^j, we obtain

$$L_\ell^j = E\big[\xi^j(k)\xi^j(k-\ell)'\big]$$
$$= \sum_{i=\ell+1}^{m} B_i^j Q^j (B_{i-\ell}^j)' + \sum_{i=\ell}^{m} G_i^j R (G_{i-\ell}^j)'. \quad (16)$$

The noise covariance matrices $Q^j = [q_{i\ell}]$ of dimension $n_v \times n_v$ and $R^j = [r_{i\ell}]$ of dimension $n_z \times n_z$ are positive definite and symmetric. By converting noise covariance matrices and the L_ℓ^j matrices as vectors as in Zhang *et al*. [32], they are related to the noise covariance identifiability matrix I^j as in (17).

$$I^j \begin{bmatrix} vec(Q^j) \\ vec(R^j) \end{bmatrix} = \begin{bmatrix} L_0^j \\ L_1^j \\ \vdots \\ L_m^j \end{bmatrix}. \quad (17)$$

As shown in [32], if matrix I^j has full column rank, then the unknown noise covariance matrices, Q^j and R^j, are uniquely identifiable. When W^j is optimal, L_ℓ^j are multiples of the innovation covariance S^j, where the scaling factor involves the minimal polynomial coefficients. For an optimal filter, it is easy to show that

$$L_\ell^j = \Big(\sum_{i=0}^{m-l} a_i^j a_{i+l}^j\Big)S^j;\ l = 0, 1, 2, ..., m. \quad (18)$$

B. Postfit Residual-Based Identifiability Conditions

Let us define $\mu^j(k)$ as the postfit residual sequence of the KF. This sequence is related to the innovation sequence $\nu^j(k)$, $k = 1, 2, ..., N$ via

$$\mu^j(k) = z(k) - H^j\hat{x}^j(k|k) = [I_{n_z} - H^j W^j(k)]\nu^j(k). \quad (19)$$

We can rewrite (19) as

$$\mu^j(k) = H^j(\tilde{F}^j)^m e^j(k-m|k-m) + \Big\{H^j\sum_{p=0}^{m-1}(\tilde{F}^j)^p$$
$$\times\big[(I_{n_x} - W^j H^j)\Gamma^j v^j(k-p-1) - W^j w^j(k-p)\big]\Big\}$$
$$+ w^j(k);\ k \geq m, \quad (20)$$

where $e^j(k+1|k+1) = \tilde{F}e^j(k|k) + (I_{n_x} - WH)\Gamma v(k) - Ww(k+1)$ is the postfit error at time $(k+1)$. Note that $\tilde{F}^j = (I_{n_x} - W^j H^j)F^j$ and $\bar{F}^j = F^j(I_{n_x} - W^j H^j)$ are similar because $\tilde{F}^j = (F^j)^{-1}\bar{F}^j F^j$.

Given the postfit residual sequence (20), let $\zeta^j(k)$ be a weighted sum of postfit residuals (see Appendix A) as,

$$\zeta^j(k) = \sum_{i=0}^{m} a_i^j \mu^j(k-i)$$
$$= \sum_{l=1}^{m} \tilde{B}_l v^j(k-l) + \sum_{l=0}^{m} \tilde{G}_l w^j(k-l), \quad (21)$$

where $\tilde{B}_l{}^j$ and $\tilde{G}_l{}^j$ are given by

$$\tilde{B}_l^j = (I_{n_z} - H^j W^j)B_l^j, \quad (22)$$

$$\tilde{G}_l^j = (I_{n_z} - H^j W^j)G_l^j. \quad (23)$$

Note that $\zeta^j(k) = (I_{n_z} - H^j W^j)\xi^j(k)$. Identifiability conditions in terms of postfit residual correlations similar to (17) ensue because $(I_{n_z} - H^j W^j)$ is invertible.

C. Output Correlations-Based Identifiability Conditions

The identifiability conditions using output correlations can be derived by using outputs only for stable open-loop systems or by using postfit residuals when the state estimation error is stabilizable when the open-loop system is unstable or marginally stable (e.g., a constant velocity target model). We will use the latter approach here.

Given (3) and (4), we can rewrite (5), the updated state estimate at time k, as

$$\hat{x}^j(k|k) = F^j\hat{x}^j(k-1|k-1) + W^j[z(k) - H^j F^j\hat{x}^j(k-1|k-1)]$$
$$= \tilde{F}^j\hat{x}^j(k-1|k-1) + W^j z(k). \quad (24)$$

We can write (19), the postfit residual sequence, as

$$\mu^j(k) = -H^j(\tilde{F}^j)^m\hat{x}^j(k-m|k-m)$$
$$- H^j\sum_{\ell=0}^{m-1}(\tilde{F}^j)^{m-1-\ell}W^j w^j(k-m+\ell+1) + z(k). \quad (25)$$

Given the postfit residual sequence (25), a weighted sum of postfit residual based on the output correlations, $\vartheta^j(k)$, can be obtained as

$$\begin{aligned}\vartheta^j(k)&=\sum_{i=0}^{m} a_i^j \mu^j(k-i)\\ &=\sum_{i=0}^{m} a_i^j\Big\{-H^j\tilde{F}^{m-i}\hat{x}^j(k-m|k-m)\\ &\quad -H^j\sum_{p=0}^{m-1-i}(\tilde{F}^j)^p W^j z(k-p)+z(k-i)\Big\}\\ &=\sum_{l=1}^{m}\hat{B}_l v^j(k-l)+\sum_{l=0}^{m}\hat{G}_l w^j(k-l)\\ &=\sum_{l=0}^{m}\hat{G}_l z(k-l). \end{aligned} \tag{26}$$

Here, the cross-covariance of $\vartheta^j(k)$ is the same as the cross-covariance of $\sum_{l=0}^{m}\hat{G}_l z(k-l)$. Identifiability conditions in terms of output correlations similar to (17) are obtained.

IV. ESTIMATING UNKNOWN FILTER PARAMETERS

A. Recursive Fading Memory-Based Innovation Correlation Estimation

We compute the sample correlation matrix $\hat{C}_{seq}^{j,k}(i)$ at sample k for model j and time lag i as a weighted combination of the correlation matrix $\hat{C}_{seq}^{j,k-1}(i)$ at the previous sample $(k-1)$ for model j and time lag i, and the samples of innovations $v^j(k-i)$ and $v^j(k)$. The tuning parameter λ, a positive constant between 0 and 1, is the weight associated with the previous sample correlation matrix. The recursive nature of the proposed algorithm makes it amenable to estimate slowly varying Q^j and R^j in nonstationary systems.

The current M sample correlation matrices at time k are used as the initial values for the next pairs of samples for recursive computation. Let us define the number of samples as N. The recursive expressions for the correlation matrices $\hat{C}_{seq}^{j,k}(i)$ are

$$\hat{C}_{seq}^{j,k}(i)=(1-\lambda)v^j(k-i)v^j(k)'+\lambda\hat{C}_{seq}^{j,k-1}(i), \tag{27}$$

$$\hat{C}_{seq}^{j,0}(i)=0;\ i=0,1,...,M-1;\ k=M,...,N. \tag{28}$$

B. Objective Function and the Gradient

The ensemble cross-correlations of a steady-state suboptimal KF are related to the closed-loop filter matrix $\bar{F}^j=F^j(I_{n_x}-W^jH^j)$, the matrix F^j, the measurement matrix H^j, the steady-state predicted covariance matrix $\bar{P}^j$, filter gain W^j, and the innovation covariance, $C^j(0)$ via [5], [21]

$$\begin{aligned}C^j(i)&=E[v^j(k)v^j(k-i)']\\ &=H^j(\bar{F}^j)^{i-1}F^j[\bar{P}^j(H^j)'-W^jC^j(0)]. \end{aligned} \tag{29}$$

The objective function Ψ^j, formulated in [32], involves minimization of the sum of normalized $C^j(i)$ with respect to the corresponding diagonal elements of $C^j(0)$ for $i>0$. The objective function is dimensionless and is zero when the filter gain is optimal and the innovation sequence is decorrelated. Formally, we can define the decorrelating objective function Ψ^j to be minimized with respect to W^j as

$$\begin{aligned}\Psi^j=&\frac{1}{2}\mathrm{tr}\Big\{\sum_{i=1}^{M-1}\big[\mathrm{diag}(C^j(0))\big]^{-\frac{1}{2}}C^j(i)'\\ &\times\big[\mathrm{diag}(C^j(0))\big]^{-1}C^j(i)\big[\mathrm{diag}(C^j(0))\big]^{-\frac{1}{2}}\Big\}, \end{aligned} \tag{30}$$

where $\mathrm{diag}(C^j)$ denotes the Hadamard product of an identity matrix with C^j. We can rewrite the objective function by substituting (29) into (30) as

$$\Psi^j=\frac{1}{2}\mathrm{tr}\Big\{\sum_{i=1}^{M-1}\phi^j(i)X^j\varphi^j(X^j)'\Big\}, \tag{31}$$

where

$$\phi^j(i)=[H^j(\bar{F}^j)^{i-1}F^j]'\varphi^j[H^j(\bar{F}^j)^{i-1}F^j], \tag{32}$$

$$X^j=\bar{P}^j(H^j)'-W^jC^j(0), \tag{33}$$

$$\varphi^j=[\mathrm{diag}(C^j(0))]^{-1}. \tag{34}$$

The gradient of objective function $\nabla_W\Psi^j$ can be computed as [32]

$$\begin{aligned}\nabla_W\Psi^j=&-\sum_{i=1}^{M-1}[H^j(\bar{F}^j)^{i-1}F^j]'\varphi^jC^j(i)\varphi^jC^j(0)-(F^j)'Z^jF^jX^j\\ &-\sum_{l=0}^{i-2}[C^j(l+1)\varphi^jC^j(i)'\varphi^jH^j(\bar{F}^j)^{i-l-2}]'. \end{aligned} \tag{35}$$

The Z^j term in (35) is computed by the Lyapunov equation.

$$\begin{aligned}Z^j=&(\bar{F}^j)'Z^j\bar{F}^j+\frac{1}{2}\sum_{i=1}^{M-1}(H^j(\bar{F}^j)^{i-1}F^j)'\varphi^jC^j(i)\varphi^jH^j\\ &+[(H^j(\bar{F}^j)^{i-1}F^j)'\varphi^jC^j(i)\varphi^jH^j]'. \end{aligned} \tag{36}$$

In computing the objective function and the gradient, we replace $C^j(i)$ by their sample estimates, $\hat{C}^{j,k}_{seq}(i)$ in (27). Evidently, the covariance estimation is a stochastic optimization problem because the cost function and the gradient depend on the realized sample paths.

C. Updating Filter Gain Sequentially

Let B be the mini-batch size and let $K = N/B$ be the number of mini-batches (we assume that N is divisible by B for simplicity). While the mini-batch gradient descent sequentially updates the M sample covariance matrices at every sample, we update the KF gain W^j when the sample index k is divisible by the size of the mini-batch B using the gradient of the objective function at sample k. Sequential mini-batch gradient descent allows more opportunities to converge to a better local minimum by frequent updates of the gain than the batch algorithm and is much less noisy than a single sample stochastic gradient algorithm [17]. Let r denote the updating index, starting with $r = 0$. The generic form of gain update is

$$(W^j)^{r+1} = (W^j)^r - (\alpha^j)^r(\nabla_W \Psi^j)^r. \tag{37}$$

The incremental gradient algorithm in (37) can be sped up by adaptively selecting the step size $(\alpha^j)^r$. Our results in [17] showed that Adam [19] and RMSProp [30] have the best accuracy and rapid convergence among all the accelerated SGD algorithms (e.g., bold driver [4], constant, subgradient [7], and Adadelta [31]) studied. Here, we show the performance results of our algorithm using the RMSProp update. RMSProp keeps track of the moving average of the squared incremental gradients for each gain element by adapting the step size elementwise.

$$\tau^j_{r,i\ell} = \gamma^j \tau^j_{r-1,i\ell} + (1-\gamma^j)[(\nabla_W \Psi^j)^r_{i\ell}]^2, \tag{38}$$

$$(\alpha^j_{i\ell})^r = \frac{(\alpha^j)^0}{\sqrt{\tau^j_{r,i\ell} + \epsilon}};\ \tau^j_0 = 0;\ (\alpha^j)^0 = \frac{c}{K}, \tag{39}$$

where $c > 0$ is a constant and K is the number of mini-batches. Here, $\gamma = 0.9$ is the default value and $\epsilon = 10^{-8}$ to prevent division by zero. When N is unknown, as in streaming data, K is absorbed into the constant c. This is not a restriction, as mini-batch size B is all we need to implement the SGD algorithm.

D. Estimation of Process and Measurement Noise Covariances

Assuming that the necessary and sufficient conditions for the identifiability of covariances are satisfied for each model [32], here we explore the noise covariance estimation using a single-pass SGD algorithm and validate it with three illustrative examples. Unlike the algorithm in [32], this algorithm is applicable to nonstationary and multiple-model systems.

From the joint covariance of the innovation sequence $\nu^j(k)$ and the postfit residual sequence $\mu^j(k)$ in (19), and the Schur determinant identity [6], [11], one can show that at the steady state (assuming constant gain, W^j and constant Q^j and R^j over large enough time intervals such that the filter achieves steady state)[32]

$$G^j = E[\mu^j(k)\mu^j(k)'] = R^j(S^j)^{-1}R^j, \tag{40}$$

where S^j is the steady-state innovation covariance. Because (40) can be interpreted as a simultaneous diagonalization problem in linear algebra [11] or as a continuous-time algebraic Riccati equation, the measurement covariance R^j can be estimated by solving the simultaneous diagonalization problem via Cholesky decomposition and eigen decomposition, or by solving a continuous-time Riccati equation as in [1], [32].

Given the estimated R^j, we can compute the process noise covariance Q^j and the steady-state updated state covariance P^j. This requires an iterative process because Q^j and P^j are coupled in the general case [32]. Let t and l denote the iteration indices starting with $t = 0$ and $l = 0$, and using an initial $(Q^j)^0 = W^j S^j W^{j\prime}$ (exact solution in the Wiener process case [32]), we initialize the steady-state updated covariance matrix P^j as the solution of the Lyapunov equation in (41)

$$\begin{aligned}(P^j)^0 =& \tilde{F}^j(P^j)^0(\tilde{F}^j)' + W^j R^j (W^j)' \\ &+ (I_{n_x} - W^j H^j)\Gamma^j (Q^j)^t (\Gamma^j)'(I_{n_x} - W^j H^j)', \end{aligned} \tag{41}$$

where $\tilde{F}^j = (I_{n_x} - W^j H^j)F^j$. We iteratively update P^j as in (42) until convergence

$$(P^j)^{l+1} = \left[\left(F^j(P^j)^l(F^j)' + \Gamma^j(Q^j)^t(\Gamma^j)'\right)^{-1} + (H^j)'(R^j)^{-1}H^j\right]^{-1}. \tag{42}$$

Given the converged P^j, Q^j will be updated in the t-loop until the estimate of Q^j converges. Proof of convergence is included in Appendix B.

$$(Q^j)^{t+1} = (\Gamma^j)^\dagger \left[(P^j + W^j S^j (W^j)' - F^j P^j (F^j)')^{t+1}\right]((\Gamma^j)')^\dagger. \tag{43}$$

V. NUMERICAL EXAMPLES

Our prior study explored the effects of varying the batch sizes and the number of observation samples for accurately estimating the unknown variance parameters in nonstationary systems [18]. Here, jumps in the estimated noise covariance for the one that is not changing may be due to the changes in the other one (for example, an estimate of R may jump when R is static but Q is changing or vice versa). This is because the Kalman gain $W = PH'(HPH' + R)^{-1}$ is impacted by both Q (through P) and R.

In this section, we explore the problem of tracking the position and velocity of an aircraft in an air traffic control (ATC) system (see Section 5.1). We also consider a three-state system for estimating the unknown

Q and R (see Section 5.2). For comparison with the algorithms in the literature, we also compare the estimation performance with noise covariance estimation algorithms in stationary systems (see Section 5.3). Finally, we explore a multiple-model scenario with a set of single-pass adaptive KFs for each mode. The multiple-model method estimates the noise covariance parameters in parallel, and then selects the probable model by the concomitant mode probability (see Section 5.4). Additional application examples may be found in [16].

In the data generation process, the system is assumed to have nonstationary noise covariance matrices. We define subgroups where each subgroup has a subset of observation samples during which the noise covariances remain constant. The noise covariances abruptly change by an unknown magnitude when one subgroup of samples ends and another starts.

Note that we present the performance of the proposed method using a single model ($J = 1$) in Sections 5.1–5.3, and then consider the multiple-model case ($J = 2$) in Section 5.4. In the estimation procedure, we set the number of burn-in samples $N_b = 50$, and the number of lags $M = 5$. All computational simulations were run on a computer with an Intel Core i7-8665U processor and 16 GB of RAM.

A. A Nonlinear ATC Scenario

We consider an ATC scenario used in [3]. The ground truth is a target moving with a constant speed of 250 m/s with an initial state specified in Cartesian coordinates. The sampling interval is $T = 1$ second. A total of 500 measurement samples were collected (500 seconds of data). The target starts a left turn of 2°/s for 30 seconds at $k = 100$, then continues straight for 70 seconds (until $k = 200$), at which time it turns right with 1°/s for 45 seconds (until $k = 245$), then left with 1°/s for 90 seconds (until $k = 335$), then right with 1°/s for 45 seconds (until $k = 380$), then continues straight for 120 seconds (until $k = 500$).

The target position measurements are generated starting from $k = 0$, and they are in polar coordinates (range r and azimuth θ) by a radar located at $[\xi_0, \eta_0] = [-10^4, 0]$, with

$$r = \sqrt{(\xi - \xi_0)^2 + (\eta - \eta_0)^2}, \tag{44}$$

$$\theta = \tan^{-1}\left(\frac{\eta - \eta_0}{\xi - \xi_0}\right), \tag{45}$$

with additive white Gaussian noise with covariance $R = \text{diag}([2500\text{m}^2, (1^\circ)^2])$. Note that the noise is added to the Cartesian converted measurements, and the true values of Q and R are used for the methods which do not estimate Q and R. For this example, we used a KF based on a second order linear kinematic model (WNA) with process noise of standard deviation 1 m/s^2 described in (46).

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \Gamma = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix}. \tag{46}$$

An interacting multiple-model (IMM) estimator with one WNA (a constant velocity model with process noise standard deviation 1 m/s^2) for the uniform motion (UM) and a nearly coordinated turn (CT) model described in (47) and (48) are used. The process noise standard deviations used in the CT model were 3 m/s^2 and 0.1°/s^2 for the UM and turn rate of the state, respectively.

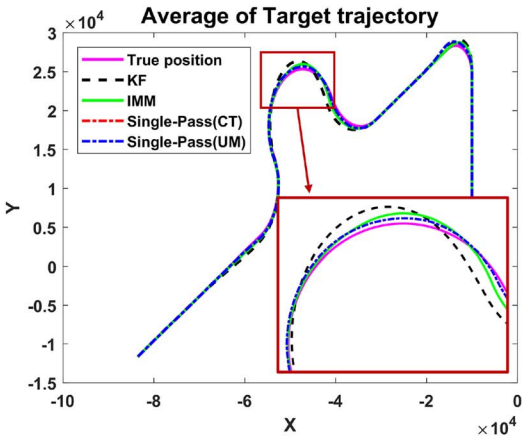
$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \Gamma = \begin{bmatrix} T^2/2 & 0 & 0 \\ T & 0 & 0 \\ 0 & T^2/2 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix}. \tag{48}$$

The mode transition probability matrix π in (49) is used for IMM estimator.

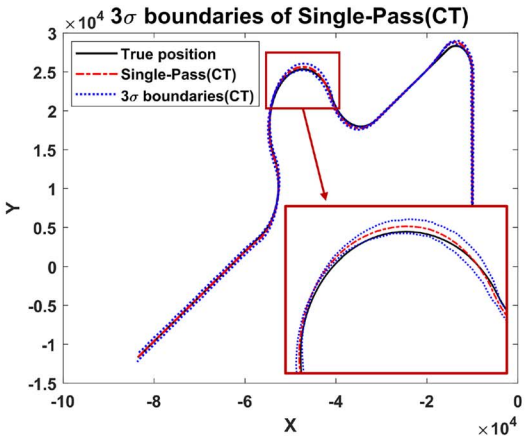
$$\pi = \begin{bmatrix} 0.95 & 0.05 \\ 0.10 & 0.90 \end{bmatrix}, \tag{49}$$

Fig. 1a shows the averaged tracking results of target motion over 100 Monte Carlo (MC) runs by KF, IMM, and the proposed method. For the single-pass SGD estimation algorithm, we considered two models using either the UM or the CT model. The proposed approach can track the target close to its true trajectory when compared to both KF and IMM. For the statistical analysis, we consider the trajectory of the single-pass SGD method with three-σ boundaries as shown in Fig. 1b. We calculate the upper and lower limits by three standard deviations (three-σ) from the mean computed over 100 MC runs. The estimates based on the CT model (even the UM model as well) are within these boundaries, which indicates that the estimates are close to the mean values. This may suggest that adapting the covariance of noise processes may overcome a lack of knowledge of the dynamics of the target to a certain extent.

$$F = \begin{bmatrix} 1 & \frac{\sin\hat{\Omega}(k)T}{\hat{\Omega}(k)} & 0 & -\frac{1-\cos\hat{\Omega}(k)T}{\hat{\Omega}(k)} & f_{\Omega,1}(k) \\ 0 & \cos\hat{\Omega}(k)T & 0 & -\sin\hat{\Omega}(k)T & f_{\Omega,2}(k) \\ 0 & \frac{1-\cos\hat{\Omega}(k)T}{\hat{\Omega}(k)} & 1 & \frac{\sin\hat{\Omega}(k)T}{\hat{\Omega}(k)} & f_{\Omega,3}(k) \\ 0 & \sin\hat{\Omega}(k)T & 0 & \cos\hat{\Omega}(k)T & f_{\Omega,4}(k) \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} f_{\Omega,1}(k) \\ f_{\Omega,2}(k) \\ f_{\Omega,3}(k) \\ f_{\Omega,4}(k) \end{bmatrix} = \begin{bmatrix} \frac{\cos\hat{\Omega}(k)T^2\hat{\dot{\xi}}(k)}{\hat{\Omega}(k)} - \frac{\sin\hat{\Omega}(k)T\hat{\dot{\xi}}(k)}{(\hat{\Omega}(k))^2} - \frac{\sin\hat{\Omega}(k)T^2\hat{\eta}(k)}{\hat{\Omega}(k)} - \frac{(-1+\cos\hat{\Omega}(k)T)\hat{\dot{\eta}}(k)}{(\hat{\Omega}(k))^2} \\ -(\sin\hat{\Omega}(k)T)T\hat{\dot{\xi}}(k) - (\cos\hat{\Omega}(k)T)T\hat{\dot{\eta}}(k) \\ \frac{\sin\hat{\Omega}(k)T^2\hat{\dot{\xi}}(k)}{\hat{\Omega}(k)} - \frac{(1-\cos\hat{\Omega}(k)T)\hat{\dot{\xi}}(k)}{(\hat{\Omega}(k))^2} + \frac{\cos\hat{\Omega}(k)T^2\hat{\dot{\eta}}(k)}{\hat{\Omega}(k)} - \frac{\sin\hat{\Omega}(k)T\hat{\dot{\eta}}(k)}{(\hat{\Omega}(k))^2} \\ (\cos\hat{\Omega}(k)T)T\hat{\dot{\xi}}(k) - (\sin\hat{\Omega}(k)T)T\hat{\dot{\eta}}(k) \end{bmatrix}. \tag{47}$$



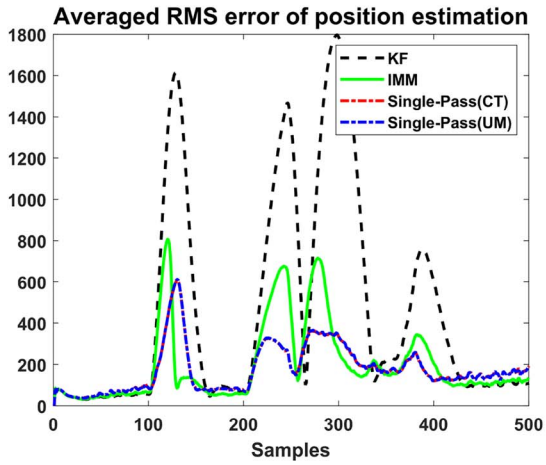
(a) Comparison of target trajectories (100 MC runs)



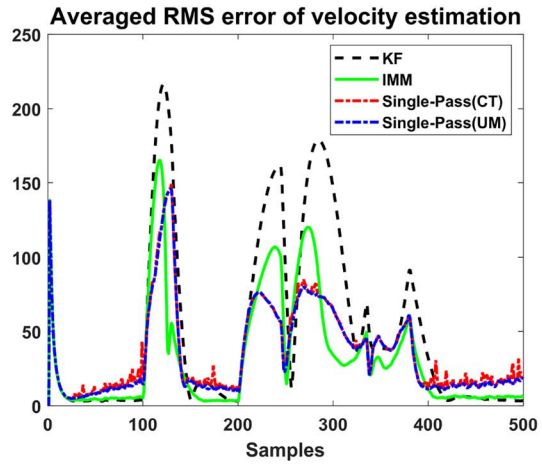
(b) Target trajectory of the proposed method with 3σ boundaries (100 MC runs)

Figure 1. Averaged tracking results of ATC motion scenario.

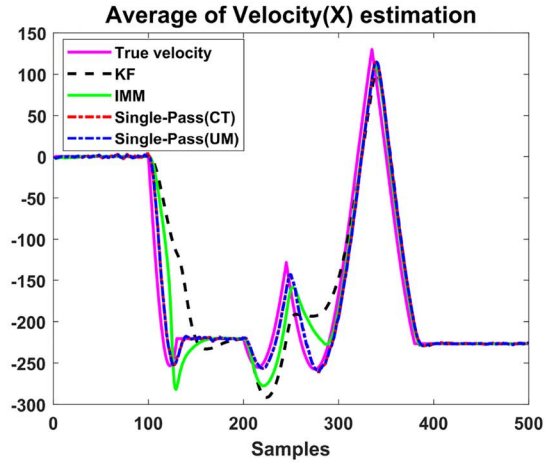
As shown in Fig. 2a, the proposed approach based on the CT model (even the UM model as well) has the peak root mean square (RMS) position error of about 200 m in the scenario considered. The proposed method reduces the RMS position error by a factor of nine when compared to a KF and by a factor of four when compared to an IMM estimator when the aircraft is maneuvering. The proposed approach shows an acceptable RMS error of velocity estimation, as shown in Fig. 2b. The proposed approach can also track the target velocity close to its true value as shown in Fig. 2c and 2d.



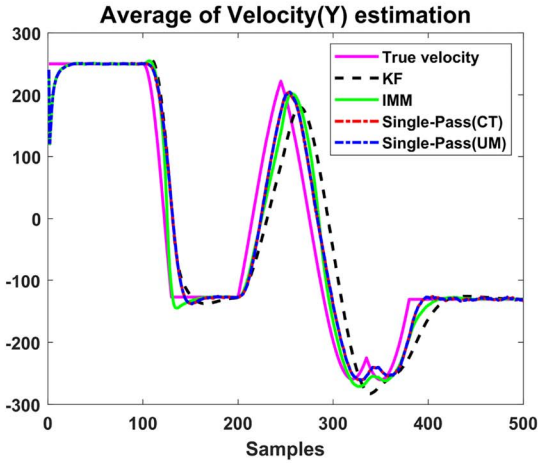
(a) RMS position estimation error



(b) RMS velocity estimation error



(c) X-velocity estimation



(d) Y-velocity estimation

Figure 2. Comparison of optimization algorithms for ATC motion estimation (100 MC runs).

Table I
Single-Pass SGD Estimation When Both Q and R Change Continuously (100 Runs; RMSProp Update)

Subgroup	R			Q			$\bar{P}_{11}$			$\bar{P}_{22}$			$\bar{P}_{33}$			W_{11}			W_{21}			W_{31}		
index	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE
1st	0.06	0.07	0.01	0.36	0.35	0.03	0.38	0.37	0.06	1.46	1.39	0.22	3.36	3.20	0.50	1.21	1.18	0.11	2.37	2.28	0.14	3.60	3.45	0.20
2nd	0.04	0.04		0.20	0.21		0.22	0.23		0.82	0.86		1.90	1.97		1.15	1.22		2.25	2.32		3.41	3.51	
3rd	0.06	0.05		0.25	0.27		0.27	0.30		1.02	1.13		2.35	2.60		1.03	1.16		2.02	2.24		3.07	3.40	
4th	0.10	0.10		0.46	0.46		0.49	0.50		1.86	1.89		4.28	4.36		1.07	1.10		2.09	2.16		3.18	3.32	
5th	0.12	0.12		0.56	0.57		0.61	0.61		2.29	2.32		5.28	5.35		1.10	1.09		2.15	2.15		3.27	3.32	

B. Scenario Where Process and Measurement Noise Covariances Change Continuously

In this scenario, we consider a three-state system using 50 000 samples where both Q and R change continuously as in the example used in [25]. The system, which has a well-conditioned observable matrix, is assumed to be as follows:

$$F = \begin{bmatrix} 0.1 & 0 & 0.1 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}, \; H = \begin{bmatrix} 0.1 & 0.2 & 0 \end{bmatrix}, \; \Gamma = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}. \tag{50}$$

The true values of Q and R are generated by first starting with piecewise constant variance values for the five subgroups of samples as $Q = [0.36, 0.20, 0.25, 0.46, 0.56]$, and $R = [0.06, 0.04, 0.06, 0.10, 0.12]$ with the values changing every 10 000 samples. The Gaussian-weighted moving average algorithm with a window size of 10,000 samples is applied to the piecewise constant noise covariances to generate smoothed continuous values.

As shown in Table I, the proposed algorithm can track the noise covariance parameters accurately when both Q and R change continuously. Because the noise covariance is changing continuously, the table values are provided only in the middle of the subgroups.

Fig. 3 shows the trajectory of noise parameters when both Q and R change continuously. Our sequential algorithm can track Q and R correctly with a smoothing weight of 0.7, and the KF is consistent when evaluated with respect to the normalized innovation squared (NIS) metric, as shown in Fig. 3c.

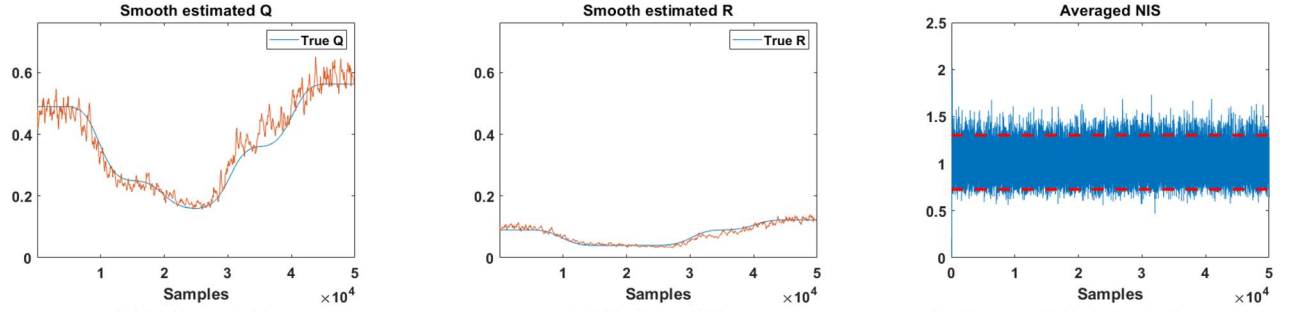
C. Comparison of Noise Covariance Estimation Algorithms on Stationary and Nonstationary Systems

Since most noise covariance estimation algorithms assume constant Q and R, in this scenario, we consider a stationary system as in the example used in [10] with 10 000 samples. We compare our single pass, multiple pass, and batch estimation algorithms with the noise covariance estimation algorithms based on the Bayesian method, the covariance matching methods (CMMs), correlation methods, and the maximum likelihood methods (MLMs). The system is assumed to be as follows:

$$F = \begin{bmatrix} 0.9 & 0 \\ -0.3 & 0.8 \end{bmatrix}, \; H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; \Gamma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{51}$$

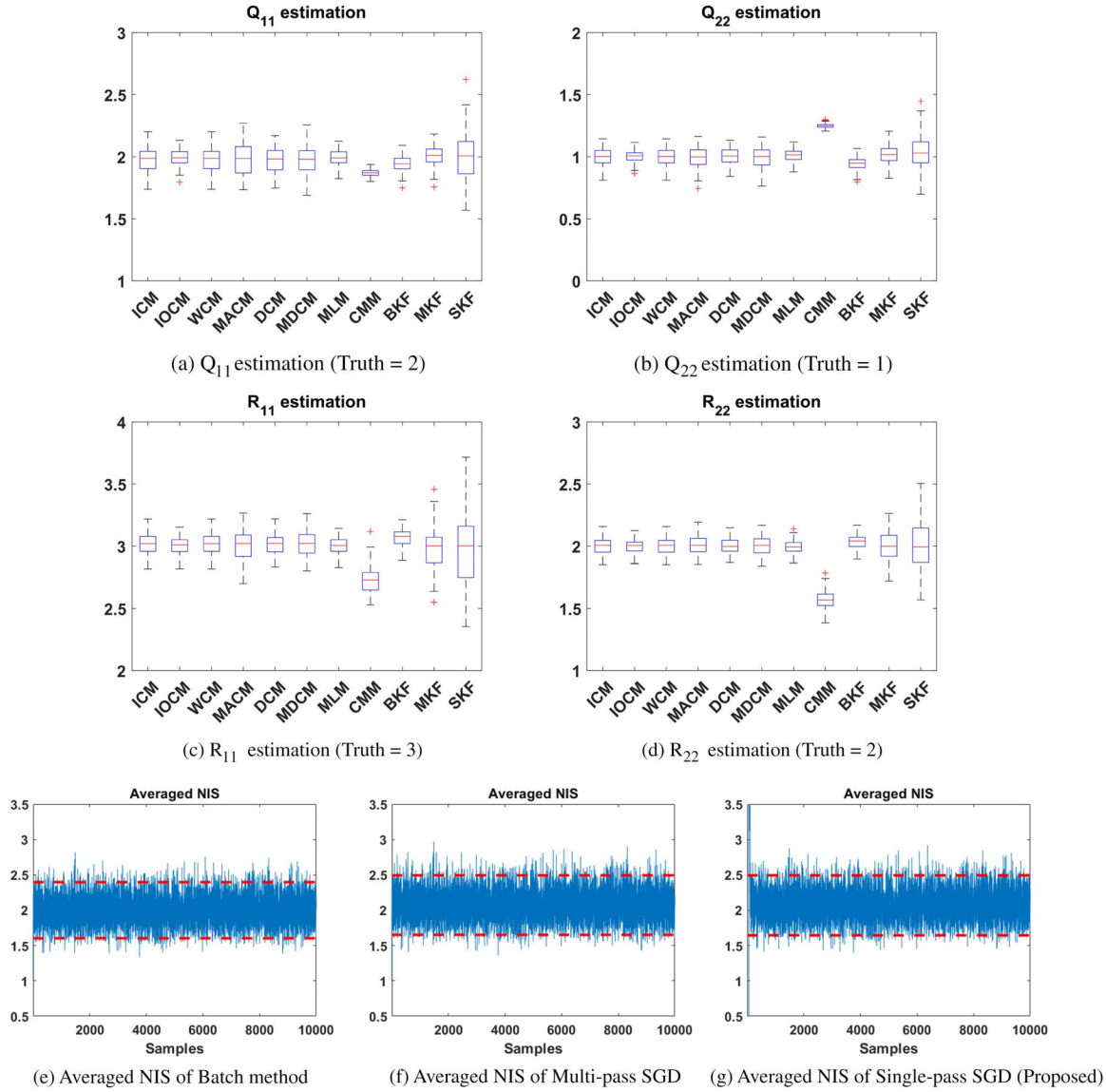
The indirect correlation method (ICM) [21], [22] relies on examining the autocovariance function (ACF) of the innovations of a linear estimator. The weighted correlation method (WCM) [5] is based on an analysis of the innovation sequence in the linear estimator, and the direct correlation method (DCM) [25] estimates the noise covariances of the innovation sequence of a stable linear estimator. In this scenario, we set an initial gain $W^0 = 0.8I_2$ for ICM, WCM, and DCM algorithms. The input–output correlation method (IOCM) [14] is designed for the linear Gaussian models by a minimization of the measurement prediction error related to an input–output model. The measurement matrix H is the identity matrix as required by IOCM algorithm, and the initial condition for estimating the coefficient matrices $\mathcal{B}^0 = O_2$.

The measurement average correlation method (MACM) [33] is based on an analysis of the covariance



(a) Estimated Q (b) Estimated R (c) Averaged NIS of Single-pass SGD

Figure 3. Trajectory of noise parameters when Q and R change continuously.



(a) Q_{11} estimation (Truth = 2)

(b) Q_{22} estimation (Truth = 1)

(c) R_{11} estimation (Truth = 3)

(d) R_{22} estimation (Truth = 2)

(e) Averaged NIS of Batch method

(f) Averaged NIS of Multi-pass SGD

(g) Averaged NIS of Single-pass SGD (Proposed)

Figure 4. Comparison of covariance estimation algorithms in stationary systems (100 MC Runs; 10 000 samples).

sequences of measurement estimate error. Two initial weight matrices $M^1_{wgt} = I_4$ and $M^2_{wgt} = I_6$ are given in this scenario. The measurement difference correlation method (MDCM) [9] directly derives the measurement estimate from other measurements without requiring state estimation. However, instead of weighting multiple measurements as MACM does, this method predicts the measurement through forward-in-time propagation of the measurement. Here, the number of measurement predictions for MDCM is set to 1. The MLM [28] relies on maximizing the likelihood function directly associated with the state space models via numerical optimization. The CMM [23] is designed for a *linear time-varying* (LTV) models with time-varying noise covariances, employing the filtering and predictive steps of a linear estimator. We set the initial noise covariances $Q^0 = I_2$ and $R^0 = I_2$ for the CMM and the single-pass SGD Kalman filter (SKF) algorithms. The batch and multipass versions of our approach (BKF and MKF) are also included for comparison purposes.

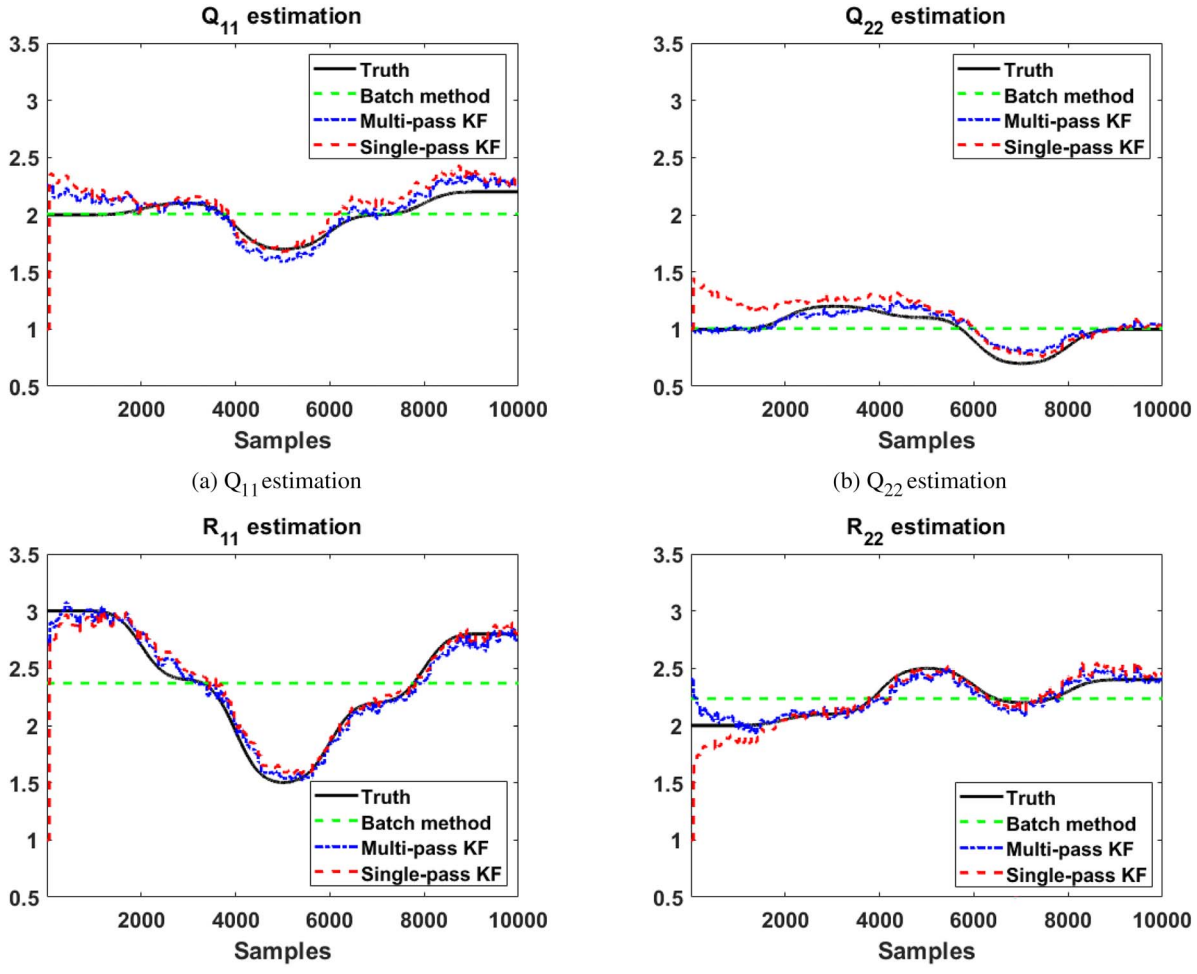
Table II shows the performance comparison of our proposed method (shown highlighted) with other algorithms for estimating noise covariances Q and R for this system averaged over 100 MC simulation runs.

Fig. 4 shows the box plots of the estimates for the various covariance estimation algorithms. Each method shows the estimates of noise covariances, with the red central mark being the median, the edges of the box being the (blue) 25th and (black) 75th percentiles, and the red crosses corresponding to the not considered outliers. Note that the batch and multipass methods estimate the parameters as well as any other method, while the single-pass SGD method estimates the noise covariances reasonably well; indeed, all filters are consistent as measured by the averaged NIS for this stationary system.

Table II
Estimates of Noise Covariances Q and R in Stationary Systems (100 MC Runs; 10 000 Samples)

Method	Q_{11} (Truth = 2)		Q_{22} (Truth = 1)		R_{11} (Truth = 3)		R_{22} (Truth = 2)	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
ICM	1.98	1.04E-02	1.00	4.72E-03	3.02	7.84E-03	2.00	4.69E-03
IOCM	1.99	4.20E-03	1.01	2.71E-03	3.01	4.68E-03	2.00	3.59E-03
WCM	1.98	1.04E-02	1.00	4.72E-03	3.02	7.84E-03	2.00	4.69E-03
MACM	1.98	1.86E-02	1.00	7.22E-03	3.01	1.49E-02	2.01	5.68E-03
DCM	1.98	9.22E-03	1.00	4.14E-03	3.02	7.26E-03	2.00	4.34E-03
MDCM	1.97	1.57E-02	1.00	6.92E-03	3.02	9.81E-03	2.00	5.51E-03
MLM	1.99	4.26E-03	1.01	2.37E-03	3.01	4.16E-03	1.99	3.23E-03
CMM	1.87	7.56E-04	1.25	3.05E-04	2.73	1.27E-02	1.57	4.80E-03
BKF	1.94	4.19E-03	0.95	2.95E-03	3.07	4.54E-03	2.04	3.55E-03
MKF	2.00	6.48E-03	1.01	5.00E-03	2.97	2.56E-02	2.00	1.39E-02
SKF	2.00	3.72E-02	1.04	1.89E-02	2.95	7.09E-02	1.99	4.01E-02

Fig. 5 shows the trajectories of noise parameters in nonstationary systems. As discussed in [18], the batch estimation method is not well-suited for nonstationary systems due to its assumption of constant noise covariances and the need for the availability of the entire observation sequence to compute both the objective function and the gradient. For this example, a change-point detection algorithm [15] is not applied to the multipass SGD method because noise covariances change continuously. The single-pass SGD method is consistent as measured by averaged NIS (not shown) and can estimate the noise covariances correctly for online streaming data.



(a) Q_{11} estimation

(b) Q_{22} estimation

(c) R_{11} estimation

(d) R_{22} estimation

Figure 5. Trajectories of noise parameters in nonstationary systems (100 MC Runs; 10 000 samples).

D. Application to Multiple-Model Cases

We investigate a situation involving an unknown motion model, utilizing a multiple-model approach. Within this algorithm, two KFs are utilized to estimate the noise covariance parameters, each tuned to a distinct model. The mode probabilities are integral to determining the active model. It is important to highlight that we regard a model as valid if its mode probability exceeds 0.66, employing the corresponding state estimate in such instances. When mode probabilities fall between 0.33 and 0.66, we combine the state estimates proportionally based on their respective posterior probabilities.

1. Scenario When Process and Measurement Noise Covariances Vary: For the multiple-model approach, we apply our proposed algorithm to the system used in [24]. Here, we assume that the observation samples are generated by Model 2, as in (53). Model 1 is assumed to be

$$F^1 = \begin{bmatrix} 1 & 1 \\ -0.1 & 0.1 \end{bmatrix},\ H^1 = \begin{bmatrix} 1 & 0 \end{bmatrix},\ \Gamma^1 = \begin{bmatrix} 1 \\ 0.4 \end{bmatrix}. \quad (52)$$

Model 2 is

$$F^2 = \begin{bmatrix} 0.8 & 1 \\ -0.4 & 0 \end{bmatrix},\ H^2 = \begin{bmatrix} 1 & 0 \end{bmatrix},\ \Gamma^2 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}. \quad (53)$$

Fig. 6 shows the trajectory of estimated parameters with 50 000 measurements, and the noise parameters change every 10,000 samples. Here, the piecewise constant functions with five subgroups for true Q and R are generated based on Q=[0.04, 0.64, 0.25, 1.00, 0.09], and $R = [0.42, 0.81, 0.49, 0.16, 0.64]$. The mode probability of the second model is higher than the first model as expected. We find that the single-pass multiple-model approach can track Q and R accurately.

2. Scenario When Measurement Noise Covariance Changes Continuously: We consider a scenario used in [26] with two dynamic models using 4000 measurement samples in which the measurement noise covariance changes continuously and compare our algorithm with the variational Bayesian method and the IMM approach.

Model 1:

$$F^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.1 \\ 0 & -0.002 & 1 \end{bmatrix},\ H^1 = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix},\ \Gamma^1 = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}. \quad (54)$$

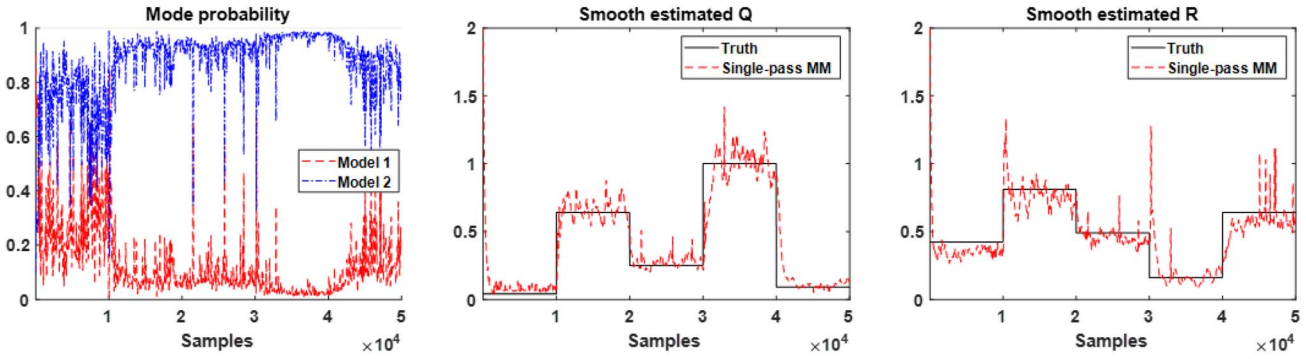
Model 2:

$$F^2 = \begin{bmatrix} 0.99 & 0 & 0 \\ 0.001 & 1 & 0.1 \\ 0 & 0 & 1 \end{bmatrix},\ H^2 = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix},\ \Gamma^2 = \begin{bmatrix} 0.01 \\ 0.05 \\ 0.1 \end{bmatrix}. \quad (55)$$

In our data generation process, we model the measurement noise variance, $R(k)$, as a continuous function in the range between 0.2 and 1 when the sampling interval, h, is 0.1 seconds, as follows:

$$R(k) = \begin{cases} 0.2 + 0.4(1 + \tanh(0.1h(k - 1000))), k \le 1500 \\ 0.2 + 0.4(1 + \tanh(0.1h(2000 - k))), \text{otherwise.} \end{cases} \quad (56)$$

Table III shows the RMSE of estimated R over 100 MC runs by the multiple-model approach. The multiple-model method estimates the noise covariance parameters in parallel, and then finds the probable model using the concomitant mode probability. Note that the measurements are generated using Model 1, as in (54). The variational bayesian adaptive kalman filter (VB-AKF) algorithm provides the best estimate when the hyper parameter $\rho = 1 - \exp(-4)$, but the RMSE value is quite sensitive to the selection of ρ. When the computation time needed for tuning the hypre parameter ρ is considered, our algorithm is superior to VB-AKF in RMSE and computational efficiency. For estimating R by the IMM filter with a multiple-model approach, each model used 111 noise models that changed uniformly between 0.1 and 1.2. The IMM filter with a large number of models (111 noise models) shows a 10% lower RMSE than our proposed method, but the IMM filter is very expensive computationally by as much as a factor of 197. Even IMM filter with 64 noise models shows slightly worse RMSE than our proposed method, but our method has better computational efficiency by a factor of 66 over the IMM.



(a) Mode probability (b) Estimated Q (c) Estimated R

Figure 6. Estimated trajectories of Q and R based on the multiple-model approach.

Table III
RMSE of Estimated R Using the Multiple-model Approach (100 MC Runs; 4000 Samples)

Method by MM	Computation time (sec)	RMSE	Description
VB-AKF	85	0.1624	$\rho = 1 - \exp(-3)$
	86	0.1178	$\rho = 1 - \exp(-4)$
	85	0.1506	$\rho = 1 - \exp(-5)$
IMM	9038	0.1009	64 noise models
	26 941	0.0836	111 noise models
Single-pass SGD	137	0.0922	RMSProp update, Batch size = 16, Smoothing weight = 0.7

Fig. 7 shows the averaged estimated trajectory of R over 100 MC runs by VB-AKF (for $\rho = 1 - \exp(-4)$), IMM (for 111 noise models) and our single-pass SGD (for RMSProp update with a mini-batch size of 16) in the multiple-model scenario. For estimating the noise covariance parameters, all optimization methods can track R correctly, but VB-AKF requires knowledge of the heuristic factor ρ, and the computation cost of IMM is substantially high. Here, the estimated trajectory of R by the single-pass algorithm was smoothed by a smoothing weight of 0.7.

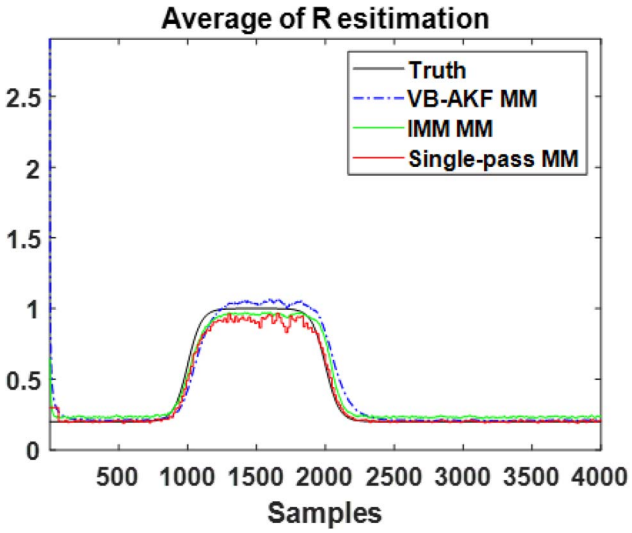


Figure 7. Trajectory of estimated R by multiple-model approach (100 MC runs).

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented a single-pass SGD algorithm that estimates the noise covariance in adaptive KFs with streaming data. Compared to the batch method or multipass sequential algorithm, our proposed streaming method is an order of magnitude faster, while still achieving acceptable root mean square error (RMSE) of the state estimates. This algorithm is suitable for nonstationary systems where noise covariances vary slowly and can occasionally exhibit abrupt changes, as well as for multiple models. The efficiency of the algorithm comes from the recursive fading memory estimation of sample cross-correlations of the innovations, along with an accelerated SGD algorithms and single-pass computations. Our proposed method has been evaluated on several test cases to demonstrate its computational efficiency, accuracy, and filter consistency when compared to extant approaches.

In the future, a number of research avenues can be pursued, including (1) estimating Q and R using one-step lag smoothed residuals; (2) automatic model selection from a library of dynamic models for model adaptation; (3) use of (16) directly in a stochastic gradient algorithm, while ensuring the positive definiteness of R and positive semi-definiteness of Q (although preliminary results are not promising); (4) use of maximum likelihood criterion instead of the normalized time correlations of innovations; and (5) explore the utility of the covariance estimation algorithm as an alternative to IMMs.

APPENDIX A. PROOF OF POSTFIT RESIDUAL-BASED IDENTIFIABILITY CONDITIONS

Let us assume the postfit error $e^j(k|k) = x(k) - \hat{x}^j(k|k)$. Note that $\tilde{F}^j = (I_{n_x} - W^j H^j)F^j$ and $\bar{F}^j = F^j(I_{n_x} - W^j H^j)$ are similar because $\tilde{F}^j = (F^j)^{-1}\bar{F}^j F^j$ given by [29], [32].

$$\begin{aligned}\mu^j(k) &= H^j(\tilde{F}^j)^m e^j(k-m|k-m) + \Big\{H^j \sum_{\ell=0}^{m-1}(\tilde{F}^j)^{m-1-\ell} \\ &\quad \times \left[(I_{n_x} - W^j H^j)\Gamma^j v^j(k-m+\ell) - W^j \right. \\ &\quad \left. \times w^j(k-m+\ell+1)\right]\Big\} + w^j(k) \\ &= H^j(\tilde{F}^j)^m e^j(k-m|k-m) + \Big\{H^j \sum_{p=0}^{m-1}(\tilde{F}^j)^p \\ &\quad \times \left[(I_{n_x} - W^j H^j)\Gamma^j v^j(k-p-1) - W^j w^j(k-p)\right]\Big\} \\ &\quad + w^j(k);\ k \geq m. \end{aligned} \tag{57}$$

The process $\zeta^j(k)$, the weighted sum of postfit residual, can be obtained as

$$\begin{aligned}\zeta^j(k) &= \sum_{i=0}^{m} a_i^j \mu^j(k-i)\\ &= \sum_{i=0}^{m} a_i^j \Big\{ H^j (\tilde{F}^j)^{m-i} e^j(k-m|k-m)\\ &\quad + H^j \sum_{p=0}^{m-1-i} (\tilde{F}^j)^p \big[(I_{n_x} - W^j H^j)\Gamma^j v^j(k-i-p-1)\\ &\quad - W^j w^j(k-i-p)\big] + w^j(k-i) \Big\}. \end{aligned} \tag{58}$$

Let $l = i + p + 1$ and use the minimal polynomial

$$\begin{aligned}\zeta^j(k) &= \sum_{i=0}^{m} a_i^j \Big\{ H^j \sum_{l=i+1}^{m} (\tilde{F}^j)^{l-i-1} \big[(I_{n_x} - W^j H^j)\\ &\quad \times \Gamma^j v^j(k-l) - W^j w^j(k-l+1)\big] + w^j(k-i) \Big\}\\ &= H^j \sum_{l=1}^{m} \sum_{i=0}^{l-1} a_i^j (\tilde{F}^j)^{l-i-1} (I_{n_z} - W^j H^j)\Gamma^j v^j(k-l)\\ &\quad - H^j \sum_{l=0}^{m-1} \sum_{i=0}^{l} a_i^j (\tilde{F}^j)^{l-i} W^j w^j(k-l) + \sum_{l=0}^{m} a_l^j w^j(k-l)\\ &= \sum_{l=1}^{m} \tilde{B}_l^j v^j(k-l) + \sum_{l=0}^{m} \tilde{G}_l^j w^j(k-l), \end{aligned} \tag{59}$$

where $\tilde{B}_l{}^j$ and $\tilde{G}_l{}^j$ are given by

$$\begin{aligned}\tilde{B}_l{}^j &= H^j \left(\sum_{i=0}^{l-1} a_i^j (\tilde{F}^j)^{l-i-1} \right) (I_{n_z} - W^j H^j)\Gamma^j\\ &= (I_{n_z} - H^j W^j) \left(\sum_{i=0}^{l-1} a_i^j (\bar{F}^j)^{l-i-1} \right) \Gamma^j\\ &= (I_{n_z} - H^j W^j) B_l^j. \end{aligned} \tag{60}$$

$$\begin{aligned}\tilde{G}_l{}^j &= -H^j \left(\sum_{i=0}^{l} a_i^j \tilde{F}^{l-i} \right) W^j + a_l^j I_{n_z}\\ &= -(I_{n_z} - H^j W^j) H^j \Big(\sum_{i=0}^{l} a_i^j (\bar{F}^j)^{l-i-1} F^j W^j + a_l^j I_{n_z}\\ &= -(I_{n_z} - H^j W^j) H^j \left(\sum_{i=0}^{l-1} a_i^j (\bar{F}^j)^{l-i-1} \right) F^j W^j\\ &\quad - a_l^j (I_{n_z} - H^j W^j) H^j (\bar{F}^j)^{-1} F^j W^j + a_l^j I_{n_z}\\ &= -(I_{n_z} - H^j W^j) H^j \left(\sum_{i=0}^{l-1} a_i^j (\bar{F}^j)^{l-i-1} \right) F^j W^j\\ &\quad - a_l^j H^j W^j + a_l^j I_{n_z}\\ &= -(I_{n_z} - H^j W^j) H^j \left(\sum_{i=0}^{l-1} a_i^j (\bar{F}^j)^{l-i-1} \right) F^j W^j\\ &\quad + a_l^j (I_{n_z} - H^j W^j)\\ &= (I_{n_z} - H^j W^j) G_l^j. \end{aligned} \tag{61}$$

APPENDIX B. PROOF OF CONVERGENCE OF THE ITERATIVE ALGORITHM FOR Q

Since the predicted error covariance is related to the updated error covariance via $\bar{P}^j = P^j + W^j S^j (W^j)'$, we have

$$P^j = F^j P^j (F^j)' + \Gamma^j Q^j (\Gamma^j)' - W^j S^j (W^j)'. \tag{62}$$

Since the Kalman gain is related to the updated error covariance via $W^j = P^j (H^j)' (R^j)^{-1}$, the first iteration of the updated state error covariance, P_1^j, can be obtained by solving the Riccati equation as

$$\begin{aligned}P_1^j &= F^j P_1^j (F^j)' + \Gamma^j Q_1^j (\Gamma^j)' - W^j S^j (W^j)'\\ &= F^j P_1^j (F^j)' + \Gamma^j Q_1^j (\Gamma^j)'\\ &\quad - P_1^j (H^j)' (R^j)^{-1} S (R^j)^{-1} H^j P_1^j\\ &= F^j P_1^j (F^j)' + \Gamma^j Q_1^j (\Gamma^j)' - P_1^j (H^j)' (G^j)^{-1} H^j P_1^j. \end{aligned} \tag{63}$$

With this solution, $W^j S^j (W^j)' \geq P_1^j (H^j)' (G^j)^{-1} H^j P_1^j$ because otherwise $P_1^j \leq 0$ at the initial iteration (Recall $\Gamma^j Q_1^j (\Gamma^j)' = W^j S^j (W^j)'$ at the initial iteration). Given $\Gamma^j Q_2^j (\Gamma^j)' = P_1^j - F^j P_1^j (F^j)' - W^j S^j (W^j)' \geq \Gamma^j Q_1^j (\Gamma^j)'$, we have the second iteration of the updated state error covariance, P_2^j, as

$$\begin{aligned}P_2^j &= F^j P_2^j (F^j)' + \Gamma^j Q_2^j (\Gamma^j)' - P_2^j (H^j)' (G^j)^{-1} H^j P_2^j\\ &= F^j P_2^j (F^j)' + P_1^j - F^j P_1^j (F^j)' - W^j S^j (W^j)'\\ &\quad - P_2^j (H^j)' (G^j)^{-1} H^j P_2^j. \end{aligned} \tag{64}$$

Evidently,

$$\begin{aligned}\delta P^j &= P_1^j - P_2^j\\ &= F^j \delta P^j (F^j)' + \Gamma^j Q_1^j (\Gamma^j)' - P_1^j + F^j P_1^j (F^j)'\\ &\quad - W^j S^j (W^j)' - P_1^j (H^j)' (G^j)^{-1} H^j P_1^j\\ &\quad + P_2^j (H^j)' (G^j)^{-1} H^j P_2^j\\ &= F^j \delta P^j (F^j)' + P_2^j (H^j)' (G^j)^{-1} H^j P_2^j - W^j S^j (W^j)'\\ &\leq 0. \end{aligned} \tag{65}$$

As the iterations proceed, $P_n^j (H^j)' (G^j)^{-1} H^j P_n^j - W^j S^j (W^j)' \to 0$, and P^j monotonically approaches P^{j*} from below.

REFERENCES

[1] W. Arnold and A. Laub
"Generalized eigenproblem algorithms and software for algebraic Riccati equations,"
Proc. IEEE, vol. 72, no. 12, pp. 1746–1754, 1984.

[2] F. Auger, M. Hilairet, J. M. Guerrero, E. Monmasson, T. Orlowska-Kowalska, and S. Katsura
"Industrial applications of the Kalman filter: A. review,"
IEEE Trans. Ind. Electron., vol. 60, no. 12, pp. 5458–5471, 2013.

[3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation with applications to tracking and navigation: Theory algorithms and software. Hoboken, NJ, USA: Wiley, 2004.

[4] R. Battiti
"Accelerated backpropagation learning: Two optimization methods,"
Complex Syst., vol. 3, no. 4, pp. 331–342, 1989.

[5] P. R. Belanger
"Estimation of noise covariance matrices for a linear time-varying stochastic process,"
Automatica, vol. 10, no. 3, pp. 267–275, 1974.

[6] D. Bertsekas
Nonlinear Programming (Athena Scientific Optimization and Computation Series). Belmont, MA, USA: Athena Scientific. 2016.

[7] S. Boyd, L. Xiao, and A. Mutapcic
Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2004(01), 2003.

[8] R. S. Bucy and P. D. Joseph
Filtering for Stochastic Processes with Applications to Guidance, vol. 326. Providence, RI, USA: American Mathematical Society. 2005.

[9] J. Duník, O. Straka, and O. Kost
Measurement difference autocovariance method for noise covariance matrices estimation.
In *Proc. IEEE 55th Conf. Decis. Control*, 2016, pp. 365–370.

[10] J. Duník, O. Straka, O. Kost, and J. Havlík
"Noise covariance matrices in state-space models: A. survey and comparison of estimation methods—Part I,"
Int. J. Adaptive Control Signal Process., vol. 31, no. 11, pp. 1505–1543, 2017.

[11] G. H. Golub and C. F. Van Loan
Matrix Computations, vol. 3. Baltimore, MD, USA: The Johns Hopkins Univ. Press. 2013.

[12] P. Hanlon and P. Maybeck
"Multiple-model adaptive estimation using a residual correlation Kalman filter bank,"
IEEE Trans. Aerosp. Electron. Syst., vol. 36, no. 2, pp. 393–406, 2000.

[13] R. E. Kalman
"A new approach to linear filtering and prediction problems,"
J. Basic Eng., vol. 82, no. 1, pp. 35–45, Mar. 1960.

[14] R. Kashyap
"Maximum likelihood identification of stochastic linear systems,"
IEEE Trans. Autom. Control, vol. 15, no. 1, pp. 25–34, 1970.

[15] R. Killick, P. Fearnhead, and I. A. Eckley
"Optimal detection of changepoints with a linear computational cost,"
J. Amer. Stat. Assoc., vol. 107, no. 500, pp. 1590–1598, 2012.

[16] H.-S Kim, A. Bienkowski, and K. R. Pattipati
"A. single-pass noise covariance estimation algorithm in multiple-model adaptive Kalman filtering. In,
Proc., 2023 IEEE Aerospace Conference, 2023.

[17] H.-S Kim, L. Zhang, A. Bienkowski, and K. R. Pattipati
"Multi-pass sequential mini-batch stochastic gradient descent algorithms for noise covariance estimation in adaptive Kalman filtering,"
IEEE Access, vol. 9, pp. 99220–99234, 2021.

[18] H.-S Kim, L. Zhang, A. Bienkowski, and K. R. Pattipati
A. single-pass noise covariance estimation algorithm in adaptive Kalman filtering for nonstationary systems.
In Proc. *IEEE 24th Int. Conf. Inf. Fusion* , 2021, pp. 556–563.

[19] D. P. Kingma and J. Ba
Adam: A. method for stochastic optimization. arXiv, 2014.

[20] X.-R Li and Y. Bar-Shalom
Multiple-model estimation with variable structure.
IEEE Trans. Autom. Control, vol. 41, no. 4, pp. 478–493, Apr. 1996.

[21] R. Mehra
"On the identification of variances and adaptive Kalman filtering,"
IEEE Trans. Autom. Control, vol. 15, no. 2, pp. 175–184, Apr. 1970.

[22] R. Mehra
"Approaches to adaptive filtering,"
IEEE Trans. Autom. Control, vol. 17, no. 5, pp. 693–698, Oct. 1972.

[23] K. Myers and B. Tapley
"Adaptive sequential estimation with unknown noise statistics,"
IEEE Trans. Autom. Control, vol. 21, no. 4, pp. 520–523, Aug. 1976.

[24] C. Neethling and P. Young
" Comments on identification of optimum filter steady-state gain for systems with unknown noise covarianc,"
IEEE Trans. Autom. Control, vol. 19, no. 5, pp. 623–625, Oct. 1974.

[25] "A. new autocovariance least-squares method for estimating noise covariances,,"
Automatica, vol. 42, no. 2, pp. 303–308, 2006.

[26] S. Sarkka and A. Nummenmaa
"Recursive noise adaptive Kalman filtering by variational bayesian approximations,"
IEEE Trans. Autom. Control, vol. 54, no. 3, pp. 596–600, Mar. 2009.

[27] S. Servadio and R. Zanetti
"Differential algebra-based multiple Gaussian particle filter for orbit determination,"
J. Optim. Theory Appl., vol. 191, no. 2, pp. 459–485, 2021.

[28] R. H. Shumway, D. S. Stoffer, and D. S. Stoffer
Time Series Analysis and Its Applications, vol 3. Berlin, Germany: Springer 2000.

[29] K. Tajima
"Estimation of steady-state kalman filter gain,"
IEEE Trans. Autom. Control, vol. 23, no. 5, pp. 944–945, 1978.

[30] T. Tieleman and G. Hinton
"Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, *COURSERA: Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

[31] M. D. Zeiler
Adadelta: An adaptive learning rate method. arXiv, 2012.

[32] L. Zhang, D. Sidoti, A. Bienkowski, K. R. Pattipati, Y. Bar-Shalom, and D. L. Kleinman
"On the identification of noise covariances and adaptive Kalman filtering: A. new look at a 50 year-old problem,"
IEEE Access, vol. 8, pp. 59362–59388, 2020.

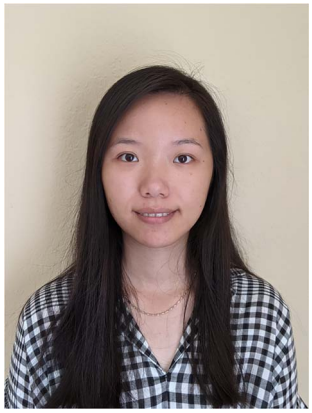
[33] J. Zhou and R. Luecke
"Estimation of the co variances of the process noise and measurement noise for a linear discrete dynamic system,"
Comput. Chem. Eng., vol. 19, no. 2, pp. 187–195, 1995.

Hee-Seung Kim received the B.S. and M.S. degrees in computer and communication engineering from the Chungbuk National University, Cheongju, South Korea, in 2011 and 2013, respectively. He is currently pursuing a Ph.D. degree in the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA.

He was a researcher at the Electronics and Telecommunications Research Institute (ETRI), Gwangju, South Korea, from 2013 to 2016. His research interests include machine learning, image processing, and adaptive Kalman filtering.

Adam Bienkowski received the B.S. and M.Eng. degrees in electrical and computer engineering from the University of Connecticut, Storrs, CT, USA, in 2013 and 2017, respectively, where he is currently pursuing a Ph.D. degree in Department of Electrical and Computer Engineering University of Connecticut, Storrs, CT, USA, under the advisement of Dr. K. R. Pattipati.

He was an Electrical Engineer at General Dynamics Electric Boat, Groton, CT, USA, from 2013 to 2017. His current research interests include modeling dynamic and uncertain environments for asset allocation and path planning, context aware decision support systems, and optimization and machine learning-based techniques for mission planning and coordination.



Lingyi Zhang received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from the University of Connecticut, Storrs, CT, USA, in 2014, 2019, and 2020, respectively. She was the co-recipient of the Tammy Blair award for best student paper at FUSION 2016. Her current research interests include modeling dynamic and uncertain environments for asset allocation and path planning, optimization-based techniques for mission planning and coordination, and adaptive Kalman filtering.

Krishna R. Pattipati received the B. Tech. degree in electrical engineering with highest honors from the Indian Institute of Technology, Kharagpur, West Bengal, India, in 1975, and the M.S. and Ph.D. degrees in systems engineering from the University of Connecticut, Storrs, CT, USA, in 1977 and 1980, respectively. He was with ALPHATECH, Inc., Burlington, USA, MA from 1980 to 1986. Since 1986, he has been with the Department of Electrical and Computer Engineering, University of Connecticut where he is currently a Distinguished Professor Emeritus and the Collins Chair Professor of Systems engineering. His research interests are in the application of systems theory, optimization and inference techniques to agile planning, anomaly detection, diagnostics, and prognostics. He has published more than 500 scholarly journal and conference papers in these areas. He is a cofounder of Qualtech Systems, Inc., a firm specializing in advanced integrated diagnostics software tools (TEAMS, TEAMS-RT, TEAMS-RDS, TEAMATE, PackNGo), and serves on the board of Aptima, Inc. Dr. Pattipati was selected by the IEEE Systems, Man, and Cybernetics (SMC) Society as the Outstanding Young Engineer of 1984 and received the Centennial Key to the Future award. He served as the *IEEE Transactions on Systems, Man, and Cybernetics Part B* from 1998 to 2001. He was co-recipient of the Andrew P. Sage Award for the Best SMC Transactions Paper for 1999, the Barry Carlton Award for the Best AES Transactions Paper for 2000, the 2002 and 2008 NASA Space Act Awards for "A Comprehensive Toolset for Model-based Health Monitoring and Diagnosis," and "Real-time Update of Fault-Test Dependencies of Dynamic Systems: A Comprehensive Toolset for Model-Based Health Monitoring and Diagnostics," the 2005 School of Engineering Outstanding Teaching Award, the 2003 AAUP Research Excellence Award at UCONN, and the 2023 Distinguished Alumnus Award from the Indian Institute of Technology, Kharagpur. He is an elected Fellow of IEEE for his contributions to discrete-optimization algorithms for large-scale systems and team decision-making, as well as a Fellow of the Connecticut Academy of Science and Engineering.

Unbiased and Consistent Electro-Optical Camera Angular Measurements With Cross-Correlated Errors and Their Fusion

J. K. Y. GOH
Y. BAR-SHALOM
R. YANG

Electro-optical (EO) camera systems are commonly used in target detection and tracking applications. Such camera systems typically comprise a suite of sensors, such as narrow/wide field of view (FOV) cameras, that provide target-originated angular measurements. To estimate the position of a point target in Cartesian space, existing techniques in literature employ the non-linear measurement mapping from the focal-plane array (FPA) to azimuth and elevation space. A common assumption made in using this conversion is that azimuth and elevation measurement errors have the same standard deviation, are uncorrelated, and are uniform across the camera's FOV. This paper presents an approach to derive the azimuth and elevation statistics, including the cross-correlation of their errors. This approach converts the raw target measurements and their covariance in the image space (FPA) to the angular space for subsequent use in Cartesian state filtering. This conversion has been validated to be unbiased and consistent, and results show that the line-of-sight (LOS) angle error variances and their correlations are in fact variable, with magnitudes dependent on the target's location in the FPA. The correct LOS angle covariance matrices should be used in Cartesian state estimation and fusion rather than the assumed constant angle variances and uncorrelated errors between the azimuth and elevation. We demonstrate a multi-sensor fusion case where the LOS angle covariance matrices of our proposed approach are used to derive the final composite target position estimate and its corresponding error covariances. The composite estimates produced from our proposed approach are proven to be statistically efficient. Compared to the use of the uncorrelated and constant LOS angle covariances, there is significant improvement to the error modeling of the fused Cartesian position covariance.

Manuscript received January 5, 2024; revised June 3, 2024; released for publication October 14, 2024

Refereeing of this contribution was handled by Stefano Coraluppi and Ramona Georgescu.

J. K. Y. Goh and R. Yang are with the DSO National Laboratories, Singapore 118225 (e-mail: gkoonyan@dso.org.sg, yrong@dso.org.sg).

Y. Bar-Shalom is with the Department of ECE, University of Connecticut, Storrs, CT 06269, USA (e-mail: yaakov.bar-shalom@uconn.edu).

1557-6418/2024/$17.00

I. INTRODUCTION

The use of electro-optical (EO) camera sensors in target detection and tracking applications has garnered much attention and intensive research in recent years. Their wide range of applications include the deployment of ground-based camera systems for aerial surveillance [12], camera sensors on-board aircraft and unmanned aerial platforms for ground target tracking [1], [6], or even for self-navigation and collision avoidance in the case of unmanned platforms [8], [9]. The many operational advantages of using camera sensors include their covertness due to them being passive, high accuracy of the angular measurements provided down to milliradians, and high data rate up to 30 Hz [7].

In video tracking, a sequence of images captures a moving target across time frames in the camera's field of view (FOV). Image processing algorithms are applied to each image to extract the centroid location of a small target in the focal-plane array (FPA) [6]. For the purpose of this paper, we focus on the conversion from FPA locations to 3D line-of-sight (LOS) angles (azimuth and elevation) in Cartesian coordinates centered at the camera. The target position and velocity can be estimated from these angular (azimuth/elevation) measurements derived from a series of coordinate transformations. Additional information is usually necessary (e.g., range or additional cameras) for the target state to be completely observable.

To track a target's position and velocity in Cartesian space, typical systems pass the angular (azimuth/elevation) measurements derived from the camera(s) and their covariances to a tracking filter. A common assumption made in the literature when using these measurements in a filter is that the azimuth and elevation measurement errors share a constant standard deviation, are uncorrelated, and are uniform across the camera's FOV. The above assumptions have been used for camera system designs that can be broadly classified into three categories: (i) a single camera sensor on a manoeuvring platform [1], [7], (ii) a camera suite comprising multiple (at least two) stationary camera sensors [11], and (iii) a sensor suite comprising a camera system with other sensor types such as radars [10]. For the sake of tracking filters performing at their best, it is essential that the measurements they use be unbiased and have correct covariances. In the case of the multisensor suite [10], data fusion is performed to derive the target's position and velocity, based on the combination of range information provided by other sensor types, such as radar, and the

highly precise angular information provided by the camera sensors.

Recent works have considered the modeling of correlation in measurement errors for camera sensor measurements, applied to the geolocation of targets. The authors in [6] formulated a camera measurement model as a nonlinear transformation function converting pixel location measurements in the image space to a 3D position in Cartesian space, based on the perspective transformation and including radial and tangential lens distortions. The covariance of the geolocation error was modeled to account for errors in pixel coordinates, intrinsic (lens distortion, skew, focal length) and extrinsic (sensor position, attitude) camera parameters, and terrain height. Results show that the geolocation covariance is representative of the actual error in the geolocation estimate. In addition, a two-camera setup to reconstruct the 3D position of the target using triangulation of the rays from corresponding image points in the left and right cameras was investigated in [4], with the total measurement error of the reconstructed point calculated using error propagation theory. The authors in [4] introduced a covariance to model the effect of the camera's orientation angle errors and the accuracy in detecting its corresponding image point.

While we recognize these approaches, the present paper's contributions are specific to camera sensors that are well calibrated with no radial and tangential distortion, and for tracking applications using angular measurements of point targets for the purpose of accurate fusion and subsequent filtering. Our approach converts the raw FPA target measurements and their covariance to the LOS angular space for subsequent use in fusion via the Maximum Likelihood approach. This conversion is shown to be unbiased and includes the variances and correlations of the measurement errors in azimuth and elevation that are in fact different at various image space (FPA) points, with magnitudes dependent on the target's location relative to the camera center.

The rest of the paper is organized as follows: Section II defines and describes the various coordinate systems used in this paper. Section III develops the conversion method from the image space to the global ENU coordinate space to derive unbiased measurements in azimuth and elevation angles, which are shown to have cross-correlated and location-dependent errors. Section IV considers the one-sensor case and presents a simulation test setup to illustrate the significance of the cross-correlated measurement errors, and proves the unbiasedness of the measurements in angular space and consistency of the derived covariance. Section V considers a two-sensor camera setup, and uses the proposed conversion approach to derive the angular error covariance matrices. The composite position measurement derived from the fusion of the cameras' LOS angles and their angular error covariances is shown to have covariance that meets the Cramer–Rao lower bound (CRLB). Lastly, Section VI presents the conclusions.

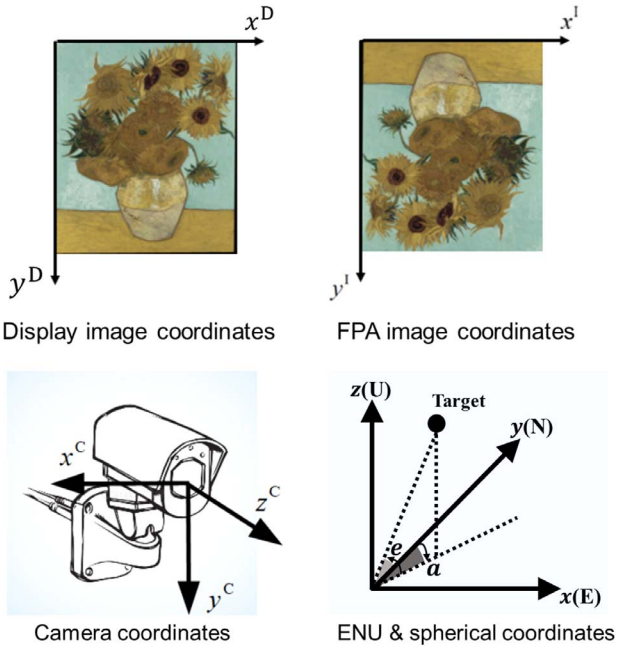


Figure 1. Display image, FPA image, camera, ENU, and spherical coordinate systems [13].

II. COORDINATE SYSTEMS

The following coordinate systems are used in this paper [13]:

- Display image coordinate system with x^D-y^D of the camera's display, shown in Fig. 1.
- Image coordinate system with x^I-y^I of the camera's FPA, shown in Fig. 1.
- Camera coordinate system with x^C-y^C-z^C centered at the camera position, shown in Fig. 1.
- Common coordinate system with x-y-z as east, north, and up (ENU) with origin at camera, with corresponding spherical coordinate system (Fig. 1) comprising:
  - azimuth (a): the clockwise angle in x(E)-y(N) plane from positive y axis to projection of target LOS onto this plane.
  - elevation (e): the angle from x(E)-y(N) plane to the target LOS.

The notations used in the paper are listed in Table I.

In practice, the flower image (on the left of Fig. 1) is shown right side up as in the camera display to correspond to the actual object. It is the resulting image after undergoing an inversion from the FPA, and the FPA image coordinates are related to the display image coordinates by

$$\mathbf{x}^I = \mathbf{h}_1\left(\mathbf{x}^D\right) = \begin{bmatrix} x^I \\ y^I \end{bmatrix} = \begin{bmatrix} P_x - x^D \\ P_y - y^D \end{bmatrix},$$
$$x^I = 0, 1, \ldots, P_x - 1, \quad y^I = 0, 1, \ldots, P_y - 1, \quad (1)$$

with P_x and P_y the (even) number of pixels in the x^I and y^I coordinates, respectively.

Table I
Notation

$\mathbf{x}^{\mathrm{A}}$	$[a \quad e]'$, the corresponding azimuth and elevation angles of a point in the common (ENU) coordinate system.
$\mathbf{x}$	$[x \quad y \quad z]'$, a point in the common (ENU) coordinate system.
$\mathbf{x}^{\mathrm{C}}$	$[x^{\mathrm{C}} \quad y^{\mathrm{C}} \quad z^{\mathrm{C}}]'$, a point in the camera coordinate system.
$\mathbf{x}^{\mathrm{I}}$	$[x^{\mathrm{I}} \quad y^{\mathrm{I}}]'$, a point in the image coordinate system in pixel units.
$\mathbf{x}^{\mathrm{D}}$	$[x^{\mathrm{D}} \quad y^{\mathrm{D}}]'$, a point in the display image coordinate system in pixel units.
$\mathbf{x}^{\mathrm{S}}$	$[x^{\mathrm{S}} \quad y^{\mathrm{S}} \quad z^{\mathrm{S}}]'$, the camera (sensor) position in ENU coordinates.
α	camera pointing azimuth or yaw (clockwise from N).
ϵ	camera pointing elevation or pitch (up from horizontal).
ρ	camera roll, clockwise around the center of the frame (ideally zero).

Next, considering the x^{C}-z^{C} plane and y^{C}-z^{C} plane given in Figs. 2 and 3, respectively, we have

$$\tan\left(\frac{\Theta_x}{2}\right) = \frac{\frac{P_x}{2}}{f} \quad \text{and} \quad \tan\left(\frac{\Theta_y}{2}\right) = \frac{\frac{P_y}{2}}{f}, \tag{2}$$

i.e.,

$$f = \frac{P_x}{2\tan\frac{\Theta_x}{2}} = \frac{P_y}{2\tan\frac{\Theta_y}{2}}, \tag{3}$$

where f is the focal length with units of measure pixel (assumed square); Θ_x and Θ_y are the FOV—angular spans—in x^{I} and y^{I}, respectively.

By similar triangles, from Fig. 2, one has

$$\frac{\frac{P_x}{2} - x^{\mathrm{I}}}{f} = \frac{x^{\mathrm{C}}}{z^{\mathrm{C}}} \quad \Rightarrow \quad x^{\mathrm{C}} = \frac{z^{\mathrm{C}}}{f}\left(\frac{P_x}{2} - x^{\mathrm{I}}\right), \tag{4}$$

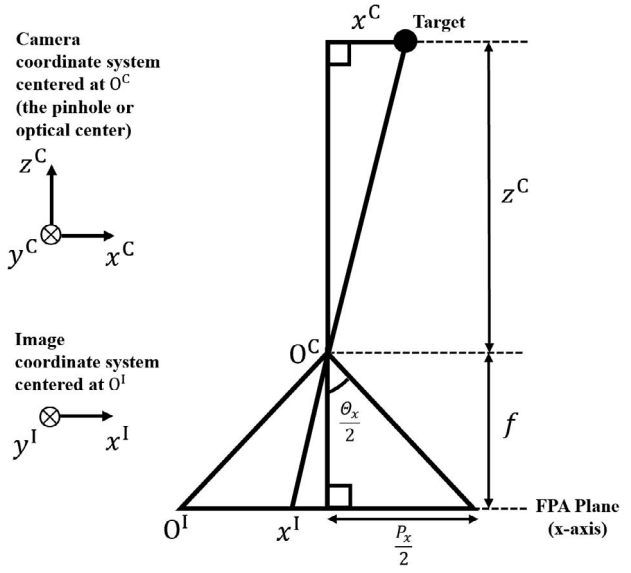


Figure 2. Diagram relating camera coordinate system to image coordinate system in x^{C}-z^{C} plane (top-down view so the FPA appears as a line); O^{C}, O^{I} denote the origins of the camera coordinate system and image coordinate system, respectively; $\otimes$ denotes the tail end of a coordinate system axis vector.

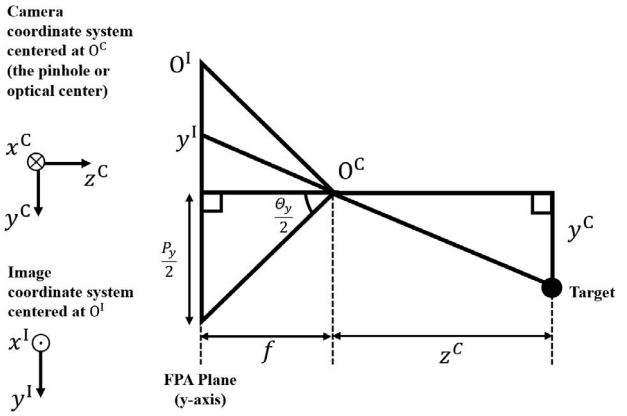


Figure 3. Diagram relating camera coordinate system to image coordinate system in y^{C}-z^{C} plane (side view so the FPA appears as a line); O^{C}, O^{I} denote the origins of the camera coordinate system and image coordinate system, respectively; $\otimes$, $\odot$ denote the tail and tip ends of a coordinate system axis vector, respectively.

and from Fig. 3,

$$\frac{\frac{P_y}{2} - y^{\mathrm{I}}}{f} = \frac{y^{\mathrm{C}}}{z^{\mathrm{C}}} \quad \Rightarrow \quad y^{C} = \frac{z^{\mathrm{C}}}{f}\left(\frac{P_y}{2} - y^{\mathrm{I}}\right). \tag{5}$$

By principle of the pinhole camera model, a point in the FPA corresponds to the projection of the 3D target point coordinates (via a line going through O^{C} as shown in Figs. 2 and 3) onto the image plane, and is the inverted image point of the target.

We further assume for convenience that

$$z^{\mathrm{C}} = f. \tag{6}$$

The LOS is then from O^{C} to the point $\left[x^{\mathrm{C}}, y^{\mathrm{C}}, z^{\mathrm{C}}\right]$ with z^{C} arbitrary since it does not affect the LOS. Then using appropriate notations, to be defined later, the coordinates of this point in the ENU system will be obtained, and they will yield the LOS angles in the ENU system.

Thus, combining equations (4)–(6), the conversion of $\mathbf{x}^{\mathrm{I}}$ to $\mathbf{x}^{\mathrm{C}}$ is given by

$$\mathbf{x}^{\mathrm{C}} = \mathbf{h}_2\left(\mathbf{x}^{\mathrm{I}}\right) = \begin{bmatrix} \frac{z^{\mathrm{C}}}{f}\left(\frac{P_x}{2} - x^{\mathrm{I}}\right) \\ \frac{z^{\mathrm{C}}}{f}\left(\frac{P_y}{2} - y^{\mathrm{I}}\right) \\ z^{\mathrm{C}} \end{bmatrix} = \begin{bmatrix} \frac{P_x}{2} - x^{\mathrm{I}} \\ \frac{P_y}{2} - y^{\mathrm{I}} \\ f \end{bmatrix}. \tag{7}$$

Next, the conversion of $\mathbf{x}^{\mathrm{C}}$ to $\mathbf{x} - \mathbf{x}^{\mathrm{S}}$ is given by

$$\begin{aligned} \mathbf{x} - \mathbf{x}^{\mathrm{S}} &= \mathbf{T}\left(\alpha, \epsilon, \rho\right)\mathbf{x}^{\mathrm{C}} \\ &= \mathbf{T}^{z^{\mathrm{C}}}(\alpha)\mathbf{T}^{x^{\mathrm{C}}}(90^{\circ} - \epsilon)\mathbf{T}^{z^{\mathrm{C}}}(-\rho)\mathbf{x}^{\mathrm{C}}, \end{aligned} \tag{8}$$

where α, ϵ, and ρ refer to the camera's yaw, pitch and roll angles, respectively, and we use the following mnemonic notations for rotations between 3D Cartesian systems

$$\mathbf{T}^{x^{\mathrm{C}}}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{9}$$

for rotation around the x^{C}-axis by ϕ from y^{C} toward z^{C}, and

$$\mathbf{T}^{z^{\mathrm{C}}}(\phi) = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{10}$$

for rotation around the z^{C}-axis by ϕ from x^{C} towards y^{C}. The rotation around the y^{C}-axis is not necessary as $\mathbf{T}^{x^{\mathrm{C}}}(90^\circ - \epsilon)$ replaces the y^{C}-axis by the z^{C}-axis, so that the rotation around the z^{C}-axis occurs twice. The combined rotation in (8) is

$$\mathbf{T}(\alpha, \epsilon, \rho) = \begin{bmatrix} s_\alpha s_\epsilon s_\rho + c_\alpha c_\rho & s_\alpha s_\epsilon c_\rho - c_\alpha s_\rho & s_\alpha c_\epsilon \\ c_\alpha s_\epsilon s_\rho - s_\alpha c_\rho & c_\alpha s_\epsilon c_\rho + s_\alpha s_\rho & c_\alpha c_\epsilon \\ -c_\epsilon s_\rho & -c_\epsilon c_\rho & s_\epsilon \end{bmatrix} \tag{11}$$

where

$$s_\alpha = \sin\alpha, \quad s_\epsilon = \sin\epsilon, \quad s_\rho = \sin\rho, \tag{12}$$

$$c_\alpha = \cos\alpha, \quad c_\epsilon = \cos\epsilon, \quad c_\rho = \cos\rho, \tag{13}$$

For a point $\mathbf{x}$ in the common ENU coordinate system, the conversion of $\mathbf{x} - \mathbf{x}^{\mathrm{S}}$ to $\mathbf{x}^{\mathrm{A}}$ is given by

$$\mathbf{x}^{\mathrm{A}} = \begin{bmatrix} a \\ e \end{bmatrix} = \mathbf{h}_3\left(\mathbf{x} - \mathbf{x}^{\mathrm{S}}\right) = \begin{bmatrix} \tan^{-1}\left(\frac{x - x^{\mathrm{S}}}{y - y^{\mathrm{S}}}\right) \\ \tan^{-1}\left(\frac{z - z^{\mathrm{S}}}{\sqrt{(x-x^{\mathrm{S}})^2 + (y-y^{\mathrm{S}})^2}}\right) \end{bmatrix}. \tag{14}$$

III. CONVERSION METHOD FOR ANGULAR MEASUREMENT ERRORS

A. Uncorrelated Measurement Error Covariance Assumption

When tracking targets' position and velocity in 3D Cartesian space, angular measurements of the target with respect to the ENU coordinate system are provided to the fuser and then to a tracking filter[1] for dynamic state estimation. The derived measurement vector from a camera sensor at time t_k is

$$\mathbf{x}^{\mathrm{A}}(t_k) = [a(t_k)\ e(t_k)]' \tag{15}$$

where $a(t_k)$ is the measured (noisy) azimuth from true North clockwise, and $e(t_k)$ is the measured elevation up from the horizontal, with reference to the camera position given by $\mathbf{x}^{\mathrm{S}}$. The corresponding measurement error covariance is typically assumed as having uncorrelated errors, i.e.,

$$\mathbf{R}^{\mathrm{A}}_{\mathrm{uncorr}} = \mathrm{diag}(\sigma_a^2, \sigma_e^2) \tag{16}$$

with σ_a and σ_e the measurement error standard deviations of azimuth and elevation, respectively. Practically, all filtering applications assume a constant measurement error standard deviation, same in both azimuth and elevation (i.e., $\sigma_a = \sigma_e$), and with uncorrelated errors between them.

B. The Measurement Error Covariance Matrix With Location Dependence and Cross-Correlation

We investigate the validity of the above assumption, whether the variances of the measurement errors are constant, and if measurement error correlation between azimuth and elevation is present. We present an approach to derive the errors of the angular measurements of the target LOS with respect to the ENU coordinate system. This involves calculating the azimuth and elevation measurements of the target (i.e., $\hat{\mathbf{x}}^{\mathrm{A}}$) from the raw target pixel measurements in the display image space (i.e., $\hat{\mathbf{x}}^{\mathrm{D}}$) through the series of transformations defined in Section II. The corresponding measurement error covariance matrix with location dependence and cross-correlated errors in azimuth-elevation space, denoted as $\mathbf{R}^{\mathrm{A}}_{\mathrm{corr}}$, is derived from the error covariance in the image space using the linearized transformation function. The validity of the first-order linearization is ascertained by confirming the unbiasedness of LOS angle errors and its covariance consistency [2], i.e., that the calculated covariance matrix is statistically compatible with the actual errors.

The target angular measurement $\hat{\mathbf{x}}^{\mathrm{A}}$ can be obtained by the global transformation function $\mathbf{h}(\cdot)$

$$\hat{\mathbf{x}}^{\mathrm{A}} = \mathbf{h}\left(\hat{\mathbf{x}}^{\mathrm{D}}\right) \triangleq \mathbf{h}_3\left[\mathbf{T}(\alpha, \epsilon, \rho)\, \mathbf{h}_2\left(\mathbf{h}_1(\hat{\mathbf{x}}^{\mathrm{D}})\right)\right], \tag{17}$$

with $\mathbf{h}_3(\cdot)$, $\mathbf{T}(\alpha, \epsilon, \rho)$, $\mathbf{h}_2(\cdot)$, and $\mathbf{h}_1(\cdot)$ defined in (14), (8), (7), and (1), respectively. Its corresponding covariance matrix (including correlations) in azimuth-elevation space is, based on first-order linearization (see, e.g., [2]), given by

$$\mathbf{R}^{\mathrm{A}}_{\mathrm{corr}} = HP^{\mathrm{D}}H' = H\begin{bmatrix} \sigma^2_{P_x} & 0 \\ 0 & \sigma^2_{P_y} \end{bmatrix} H', \tag{18}$$

where σ_{P_x} and σ_{P_y} are the measurement error standard deviations (in pixels) in the x^{D} and y^{D} axes, respectively[2], and H is the global linearized function

$$H = H_3 H_{\mathbf{T}(\alpha,\epsilon,\rho)} H_2 H_1. \tag{19}$$

as a composite of the Jacobians of the functions $\mathbf{h}_3(\cdot)$, $\mathbf{T}(\alpha, \epsilon, \rho)$, $\mathbf{h}_2(\cdot)$, and $\mathbf{h}_1(\cdot)$ defined in (14), (8), (7), and (1), respectively. Specifically,

$$\begin{aligned} H_3 = \frac{\partial \mathbf{x}^{\mathrm{A}}}{\partial(\mathbf{x} - \mathbf{x}^{\mathrm{S}})} &= \begin{bmatrix} \frac{\partial a}{\partial(x-x^{\mathrm{S}})} & \frac{\partial a}{\partial(y-y^{\mathrm{S}})} & \frac{\partial a}{\partial(z-z^{\mathrm{S}})} \\ \frac{\partial e}{\partial(x-x^{\mathrm{S}})} & \frac{\partial e}{\partial(y-y^{\mathrm{S}})} & \frac{\partial e}{\partial(z-z^{\mathrm{S}})} \end{bmatrix} \\ &= \begin{bmatrix} \frac{y-y^{\mathrm{s}}}{r_{xy}^2} & -\frac{x-x^{\mathrm{s}}}{r_{xy}^2} & 0 \\ -\frac{(x-x^{\mathrm{s}})(z-z^{\mathrm{s}})}{r_{xy}r^2} & -\frac{(y-y^{\mathrm{s}})(z-z^{\mathrm{s}})}{r_{xy}r^2} & \frac{r_{xy}}{r^2} \end{bmatrix}, \end{aligned} \tag{20}$$

where

$$r_{xy} \triangleq \sqrt{(x - x^{\mathrm{s}})^2 + (y - y^{\mathrm{s}})^2} \tag{21}$$

[1] See [3] for the various configurations in multisensor tracking.

[2] These are the same as the errors in (inverted image coordinates) x^{I}, y^{I}.

$$r \triangleq \sqrt{(x-x^{\mathrm{S}})^2+(y-y^{\mathrm{S}})^2+(z-z^{\mathrm{S}})^2}, \qquad (22)$$

and

$$H_{\mathbf{T}(\alpha,\epsilon,\rho)} = \frac{\partial\left(\mathbf{x}-\mathbf{x}^{\mathrm{S}}\right)}{\partial \mathbf{x}^{\mathrm{C}}} = \begin{bmatrix} \frac{\partial(x-x^{\mathrm{S}})}{dx^{\mathrm{C}}} & \frac{\partial(x-x^{\mathrm{S}})}{dy^{\mathrm{C}}} & \frac{\partial(x-x^{\mathrm{S}})}{dz^{\mathrm{C}}} \\ \frac{\partial(y-y^{\mathrm{S}})}{dx^{\mathrm{C}}} & \frac{\partial(y-y^{\mathrm{S}})}{dy^{\mathrm{C}}} & \frac{\partial(y-y^{\mathrm{S}})}{dz^{\mathrm{C}}} \\ \frac{\partial(z-z^{\mathrm{S}})}{dx^{\mathrm{C}}} & \frac{\partial(z-z^{\mathrm{S}})}{dy^{\mathrm{C}}} & \frac{\partial(z-z^{\mathrm{S}})}{dz^{\mathrm{C}}} \end{bmatrix} = \mathbf{T}(\alpha,\epsilon,\rho), \qquad (23)$$

as given in (11), and

$$H_2 = \frac{\partial \mathbf{x}^{\mathrm{C}}}{\partial \mathbf{x}^{\mathrm{I}}} = \begin{bmatrix} \frac{\partial x^{\mathrm{C}}}{\partial x^{\mathrm{I}}} & \frac{\partial x^{\mathrm{C}}}{\partial y^{\mathrm{I}}} \\ \frac{\partial y^{\mathrm{C}}}{\partial x^{\mathrm{I}}} & \frac{\partial y^{\mathrm{C}}}{\partial y^{\mathrm{I}}} \\ \frac{\partial z^{\mathrm{C}}}{\partial x^{\mathrm{I}}} & \frac{\partial z^{\mathrm{C}}}{\partial y^{\mathrm{I}}} \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix} \qquad (24)$$

and

$$H_1 = \frac{\partial \mathbf{x}^{\mathrm{I}}}{\partial \mathbf{x}^{\mathrm{D}}} = \begin{bmatrix} \frac{\partial x^{\mathrm{I}}}{\partial x^{\mathrm{D}}} & \frac{\partial x^{\mathrm{I}}}{\partial y^{\mathrm{D}}} \\ \frac{\partial y^{\mathrm{I}}}{\partial x^{\mathrm{D}}} & \frac{\partial y^{\mathrm{I}}}{\partial y^{\mathrm{D}}} \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}. \qquad (25)$$

Note the location dependence of the elements of (20).

IV. ONE-SENSOR CASE: SIMULATION RESULTS

Following the conversion (17) and (18), we evaluate the angular measurement error covariance matrix. Next we present the scenarios considered to illustrate the location dependence and the correlation of the errors at various points in the FPA, and the tests whether the conversion method is unbiased and has a consistent covariance [2].

A. Simulation Scenario

We consider a camera positioned at the origin in global 3D Cartesian space, i.e., $\mathbf{x}^{\mathrm{S}} = [0,0,0]'$, with the orientation angles of the camera $(\alpha,\epsilon,\rho) = (0,0,0)$. This is for the sake of illustration; for other orientations, (23) will differ. The aspect ratio of the camera is 16:9, which is typical of commercial camera sensors. This translates to a horizontal and vertical FOV of $\Theta_x = 60^\circ$ and $\Theta_y = 37.5^\circ$, respectively, and an image with $P_x = 1920$ pixels by $P_y = 1080$ pixels for a 2 megapixel (MP) camera, and $P_x = 3840$ pixels by $P_y = 2160$ pixels for a 8MP camera. The camera records measurements when the target is in its FOV with FPA measurement error standard deviations $\sigma_{P_x} = \sigma_{P_y} = 1$ pixel[3]. This error is equivalent to angles of 0.031° (or 0.541 mrad) and 0.016° (or 0.279 mrad) on average for a 2MP and 8MP camera, respectively.

[3] As shown in [5] using a physics-based model, the optimal measurement extractor for a point target can reach a s.d. of $\frac{1}{2}$ pixel. We took a more conservative approach here.

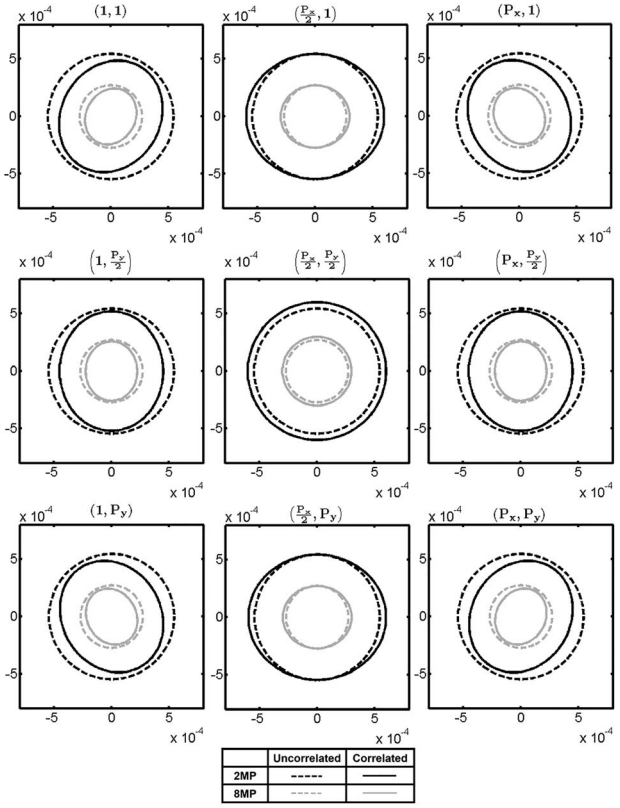


Figure 4. Azimuth-elevation error ellipse plots with assumed uncorrelated errors (dotted) and actual correlated errors (solid) at each of the chosen image points $(x_i^{\mathrm{D}}, y_i^{\mathrm{D}})$, $i = 1,\ldots,9$, from exhaustive combination pairs of $x_i^{\mathrm{D}} \in \{1, \frac{P_x}{2}, P_x\}$ and $y_i^{\mathrm{D}} \in \{1, \frac{P_y}{2}, P_y\}$, for a 2MP camera (black) and 8MP camera (gray). The locations are according to display image coordinates in Fig. 1.

B. Significance of Correlation Errors

For the given camera specifications, Fig. 4 shows measurement error ellipses in azimuth-elevation that were generated for each of nine selected points in the image plane spanning left to right of x^{D} axis and top to bottom of y^{D} axis. The nine selected points of the display image plane include: (i) top-left $(1,1)$, (ii) top-center $\left(\frac{P_x}{2},1\right)$, (iii) top-right $(P_x,1)$, (iv) mid-left $\left(1,\frac{P_y}{2}\right)$, (v) center $\left(\frac{P_x}{2},\frac{P_y}{2}\right)$, (vi) mid-right $\left(P_x,\frac{P_y}{2}\right)$, (vii) bottom-left $(1,P_y)$, (viii) bottom-center $\left(\frac{P_x}{2},P_y\right)$, and (ix) bottom-right (P_x,P_y). A baseline uncorrelated measurement error circle is also shown (in dotted line) to represent the uncorrelated error covariance $\mathbf{R}^{\mathrm{A}}_{\mathrm{uncorr}}$, with $\sigma_a = \sigma_e = \sigma_{P_x}\frac{\Theta_x}{P_x}$, for comparison with the correlated and location-dependent measurement error ellipses $\mathbf{R}^{\mathrm{A}}_{\mathrm{corr}}$ (in solid line) derived from the proposed conversion method. Error ellipses of the 2MP and 8MP camera are colored black andgray, respectively.

To determine if the measurement correlation error is significant, the percentage difference between the areas of the correlated and uncorrelated error ellipses is

Table II
Percentage Difference Between Areas of Correlated Error Ellipses and Uncorrelated Error Circles at Each of Nine Image Points $(x^{\mathrm{D}}, y^{\mathrm{D}})$ for 2MP and 8MP Camera

MP	y^{D} \ x^{D}	1	$\frac{P_x}{2}$	P_x
2	1	−26.8%	10.0%	−26.8%
	$\frac{P_y}{2}$	−21.0%	21.6%	−21.0%
	P_y	−26.8%	10.0%	−26.8%
8	1	−26.8%	10.0%	−26.8%
	$\frac{P_y}{2}$	−21.0%	21.6%	−21.0%
	P_y	−26.8%	10.0%	−26.8%

calculated as follows:

$$\mathrm{d} = \frac{\pi\left(\sqrt{\det\left(\mathbf{R}^{\mathrm{A}}_{\mathrm{corr}}\right)} - \sqrt{\det\left(\mathbf{R}^{\mathrm{A}}_{\mathrm{uncorr}}\right)}\right)}{\pi\sqrt{\det\left(\mathbf{R}^{\mathrm{A}}_{\mathrm{uncorr}}\right)}} \times 100\%, \quad (26)$$

where $\det(\mathbf{A})$ is the determinant of matrix $\mathbf{A}$. A magnitude of $|\mathrm{d}| \geq 10\%$ will imply that the correlated measurement error is significant. A positive percentage difference value implies the error ellipse is larger in area than the baseline uncorrelated error circle, while a negative percentage difference value implies the error ellipse is smaller in area than the baseline uncorrelated error circle. Table II shows the percentage difference between the areas of the correlated and uncorrelated error ellipses for each of the nine selected image points.

With reference to Fig. 4, the error ellipses of the 8MP camera (in gray) are 4 times smaller in area than those of the 2MP camera (in black), due to the larger number of pixels in the image space for the 8MP camera. The ellipses' shapes and orientations of both correlated and uncorrelated error covariances are similar for both 2MP and 8MP cameras, due to both cameras sharing the same aspect ratio. From Table II, the percentage difference values are the same for both cases of the 2MP and 8MP cameras. Given the similarities in observations in both Fig. 4 and Table II, we can, for simplicity, consider the results (of both figure and result table) of either camera for subsequent analysis and discussion in the rest of the paper.

From Table II, since all percentage difference values have magnitude greater than or equal to 10%, the correlations of azimuth and elevation measurement errors are considered significant at all nine selected points of the image.

The most significant differences are at the corner points of the image, where the correct ellipse (with the error correlations) is 26.8% smaller in area than the circles corresponding to the (assumed) uncorrelated errors at these points. The next most significant difference is at the center point of the image, with the correct error ellipse area 21.6% larger than the (assumed) uncorrelated error circle at this point. This is followed by the mid-left and mid-right points of the image, with the correlated error ellipse 21.0% larger in area than the uncorrelated error circle at this point. The least significant difference is at the top-center and bottom-center points of the image, with the correlated error ellipse 10.0% larger in area than the uncorrelated error circle at these points.

C. Unbiasedness and Covariance Consistency

We next assess whether the proposed conversion method is unbiased and that the derived covariance matrix $\mathbf{R}^{\mathrm{A}}_{\mathrm{corr}}$ is consistent.

As the global transformation function $\mathbf{h}(\cdot)$ defined in (17) mapping points in the image space to azimuth-elevation space is nonlinear, a set of $N = 10\,000$ random points with Gaussian noise was generated for each selected image point modeling the probability density distributions in image space, i.e., for each of nine selected image points, separate realizations of N points were generated according to the distribution

$$\left\{\mathbf{x}^{\mathrm{D}}_{i,k}\right\}_{k=1}^{N} = \left\{\begin{bmatrix} x^{\mathrm{D}}_{i,k} \\ y^{\mathrm{D}}_{i,k} \end{bmatrix}\right\}_{k=1}^{N} \sim \mathcal{N}\left(\begin{bmatrix} x^{\mathrm{D}}_{i} \\ y^{\mathrm{D}}_{i} \end{bmatrix}, \begin{bmatrix} \sigma^2_{P_x} & 0 \\ 0 & \sigma^2_{P_y} \end{bmatrix}\right) \quad i = 1, \ldots, 9, \quad (27)$$

where k is the index of the Monte–Carlo runs. Each set of points undergoes the non-linear transformation $\mathbf{h}(\cdot)$ to derive a distribution of points

$$\left\{\mathbf{x}^{\mathrm{A}}_{i,k}\right\}_{k=1}^{N} = \left\{\begin{bmatrix} a_{i,k} \\ e_{i,k} \end{bmatrix}\right\}_{k=1}^{N} \qquad i = 1, \ldots, 9 \quad (28)$$

in azimuth-elevation space corresponding to the ENU coordinate system. The set (28) is based on FPA random points specified by (27). The sample mean of the angles' distribution is

$$\bar{\mathbf{x}}^{\mathrm{A}}_{i} = \begin{bmatrix} \bar{a}^{\mathrm{A}}_{i} \\ \bar{e}^{\mathrm{A}}_{i} \end{bmatrix} = \frac{1}{N}\sum_{k=1}^{N} \mathbf{x}^{\mathrm{A}}_{i,k} \qquad i = 1, \ldots, 9, \quad (29)$$

and the sample covariance is

$$\bar{\mathbf{R}}^{\mathrm{A}}_{\mathrm{corr},i} = \frac{1}{N}\sum_{k=1}^{N}\left(\mathbf{x}^{\mathrm{A}}_{i,k} - \bar{\mathbf{x}}^{\mathrm{A}}_{i}\right)\left(\mathbf{x}^{\mathrm{A}}_{i,k} - \bar{\mathbf{x}}^{\mathrm{A}}_{i}\right)' \quad i = 1, \ldots, 9. \quad (30)$$

Next, each of the nine selected image points $\mathbf{x}^{\mathrm{I}}_{i}$, $i = 1, \ldots, 9$ undergoes the global transformation function $\mathbf{h}(\cdot)$ defined in (17) to obtain the converted angular state $\hat{\mathbf{x}}^{\mathrm{A}}_{i}$ comprising the LOS angles, $\hat{a}^{\mathrm{A}}_{i}$ and $\hat{e}^{\mathrm{A}}_{i}$.

The ratio of the bias error to the baseline measurement error standard deviation is calculated in the azimuth and elevation components separately

$$r_{a,i} = \frac{\hat{a}^{\mathrm{A}}_{i} - \bar{a}^{\mathrm{A}}_{i}}{\sigma_{a,i}} \quad \text{and} \quad r_{e,i} = \frac{\hat{e}^{\mathrm{A}}_{i} - \bar{e}^{\mathrm{A}}_{i}}{\sigma_{e,i}}, \quad (31)$$

where $\sigma_{a,i}$ and $\sigma_{e,i}$ are the square roots of the diagonal terms of (30). These are $\mathcal{N}\left(0, \frac{1}{N}\right)$ distributed (see, e.g., [2]), so they have to be in the 95% region: $\left[-\frac{2}{\sqrt{N}}, \frac{2}{\sqrt{N}}\right] =$

Table III
Ratio r_a of the Bias to the Baseline Measurement Error Standard Deviation for Azimuth Component of 2MP and 8MP Cameras

MP	y^{D} \ x^{D}	1	$\frac{P_x}{2}$	P_x
2	1	0.0101	−0.0022	0.0051
	$\frac{P_y}{2}$	0.0063	0.0048	−0.0051
	P_y	0.0106	−0.0063	0.0032
8	1	−0.0004	0.0036	−0.0022
	$\frac{P_y}{2}$	−0.0106	−0.0142	0.0116
	P_y	0.0008	−0.0066	0.0111

Table IV
Ratio r_e of the Bias to the Baseline Measurement Error Standard Deviation for Elevation Component of 2MP and 8MP Cameras

MP	y^{D} \ x^{D}	1	$\frac{P_x}{2}$	P_x
2	1	0.0051	−0.0147	0.0062
	$\frac{P_y}{2}$	−0.0092	0.0160	0.0038
	P_y	−0.0009	0.0064	−0.0013
8	1	0.0025	−0.0128	0.0054
	$\frac{P_y}{2}$	0.0138	−0.0137	−0.0060
	P_y	−0.0048	0.0124	0.0081

$[-0.02, 0.02]$ to confirm the unbiasedness of the proposed conversion method. Tables III and IV show the results of the ratio for the azimuth (r_a) and elevation (r_e) components, respectively, for both the 2MP and 8MP cameras considered. It can be seen that all ratio values are within the 95% region of ± 0.02. Thus, the proposed conversion method is unbiased.

Lastly, the following covariance consistency test [2] is used:

$$\kappa_i \triangleq \frac{1}{N}\sum_{k=1}^{N}\left(\mathbf{x}_{i,k}^{\mathrm{A}}-\hat{\mathbf{x}}_i^{\mathrm{A}}\right)'\left(\bar{\mathbf{R}}_{\mathrm{corr},i}^{\mathrm{A}}\right)^{-1}\left(\mathbf{x}_{i,k}^{\mathrm{A}}-\hat{\mathbf{x}}_i^{\mathrm{A}}\right) \leq \frac{1}{N}\chi_{2N}^2(1-\alpha) \quad i=1,\ldots,9, \tag{32}$$

where $\chi_{2N}^2(1-\alpha)$ is α-tail probability of the chi-square distribution with $2N$ degrees of freedom. If satisfied, these tests confirm the unbiasedness of the conversion and the correctness of the covariance $\mathbf{R}_{\mathrm{corr},i}^{\mathrm{A}}$, including the correlation between azimuth and elevation errors, for each of the nine selected image points [2]. Table V shows the results of the consistency test values κ_i for both the 2MP and 8MP cameras considered. It can be seen that that all covariance test statistics are below the $1-\alpha = 95\%$ bound of $\frac{1}{10\,000}\chi_{20\,000}^2(0.95) = 2.0330$.

Thus, the proposed conversion method that yields location-dependent variances and correlations is unbiased and consistent.

Table V
Test for Covariance Matrix Consistency of 2MP and 8MP Cameras

MP	y^{D} \ x^{D}	1	$\frac{P_x}{2}$	P_x
2	1	2.0002	2.0002	2.0001
	$\frac{P_y}{2}$	2.0001	2.0002	2.0001
	P_y	2.0002	2.0001	2.0000
8	1	2.0000	2.0002	2.0000
	$\frac{P_y}{2}$	2.0004	2.0003	2.0002
	P_y	2.0000	2.0002	2.0003

V. TWO-SENSOR CASE: MULTI-SENSOR FUSION AND SIMULATION RESULTS

In this section, we derive the composite target position measurement in 3D Cartesian space, by fusing LOS, i.e., 2D, target angle measurements of a two-camera setup. This derivation can be divided into two stages:

Stage 1. Apply our proposed conversion method in Section III-B to derive, for each camera, the target LOS angle measurements with their corresponding angular error covariances with location-dependent variances and cross-correlated errors.

Stage 2. The maximum likelihood (ML) estimate of the 3D composite target position measurement is obtained from fusing[4] the LOS angle measurements of the two cameras using a numerical search performed via the iterated least squares (ILS) technique [2]. The derived ML estimate is proven to be statistically efficient, and as such, the covariance matrix from the CRLB can be used as the measurement error covariance matrix for the resulting composite target 3D position measurement [11] in subsequent track filtering.

It is envisioned that the more appropriate angular error covariance of our approach in Stage 1, compared to an uncorrelated angular error covariance baseline, would translate to improvements to the final composite target position measurement and its Cartesian measurement error covariance matrix in Stage 2.

A. Derivation of Composite (Fused) Position Measurement and Its Corresponding Cartesian Measurement Error Covariance

Consider 2 camera sensors with known sensor positions $\mathbf{x}^{\mathrm{s}_j} = [x^{\mathrm{s}_j}, y^{\mathrm{s}_j}, z^{\mathrm{s}_j}]'$, $j = 1, 2$.

Stage 1. Following the conversion (17), (18), we have the sensor-specific (s_j) LOS measurements and their corresponding covariances denoted $\mathbf{x}_{\mathrm{s}_j}^{\mathrm{A}}$ and $\mathbf{R}_{\mathrm{s}_j,\mathrm{corr}}^{\mathrm{A}}$, $j = 1, 2$, respectively.

Stage 2. The LOS measurements to the target from Stage 1 are fused to obtain a composite position measurement (estimate) $\hat{\mathbf{x}}^{\mathrm{F}} = \left[x^{\mathrm{F}}, y^{\mathrm{F}}, z^{\mathrm{F}}\right]'$ in ENU space [11].

[4]This is Type 3 fusion [3] with the two cameras assumed synchronized.

The measurement $\mathbf{x}^{\mathrm{A}}_{\mathrm{s}_j}$ from sensor s_j relates to the 3D position according to

$$\mathbf{x}^{\mathrm{A}}_{\mathrm{s}_j} = \mathbf{g}_{\mathrm{s}_j}\left(\mathbf{x}^{\mathrm{F}}, \mathbf{x}^{\mathrm{s}_j}\right) + \mathbf{w}_{\mathrm{s}_j}, \qquad j = 1, 2, \tag{33}$$

where

$$\mathbf{w}_{\mathrm{s}_j} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}^{\mathrm{A}}_{\mathrm{s}_j,\mathrm{corr}}) \tag{34}$$

are the angle measurement noises, and

$$\mathbf{g}_{\mathrm{s}_j}\left(\mathbf{x}^{\mathrm{F}}, \mathbf{x}^{\mathrm{s}_j}\right) = \mathbf{h}_3\left(\mathbf{x}^{\mathrm{F}} - \mathbf{x}^{\mathrm{s}_j}\right) \tag{35}$$

with $\mathbf{x}^{\mathrm{A}} = \mathbf{x}^{\mathrm{A}}_{\mathrm{s}_j}$, $\mathbf{x} = \mathbf{x}^{\mathrm{F}}$ and $\mathbf{x}^{\mathrm{S}} = \mathbf{x}^{\mathrm{s}_j}$ in (14).

The composite position measurement $\hat{\mathbf{x}}^{\mathrm{F}}$ in ENU space is obtained by maximizing the likelihood function of $\mathbf{x}^{\mathrm{F}}$ (based on $\mathbf{x}^{\mathrm{A}}_{\mathrm{s}_1}, \mathbf{x}^{\mathrm{A}}_{\mathrm{s}_2}$, not shown for simplicity)

$$\mathbf{\Lambda}\left(\mathbf{x}^{\mathrm{F}}\right) = \prod_{j=1}^{2} p(\mathbf{x}^{\mathrm{A}}_{\mathrm{s}_j}|\mathbf{x}^{\mathrm{F}}), \tag{36}$$

where

$$p(\mathbf{x}^{\mathrm{A}}_{\mathrm{s}_j}|\mathbf{x}^{\mathrm{F}}) = |2\pi\mathbf{R}^{\mathrm{A}}_{\mathrm{s}_j,\mathrm{corr}}|^{-\frac{1}{2}} \cdot \exp\Big(-\frac{1}{2}\left[\mathbf{x}^{\mathrm{A}}_{\mathrm{s}_j} - \mathbf{g}_{\mathrm{s}_j}\left(\mathbf{x}^{\mathrm{F}}, \mathbf{x}^{\mathrm{s}_j}\right)\right]' \left(\mathbf{R}^{\mathrm{A}}_{\mathrm{s}_j,\mathrm{corr}}\right)^{-1}\left[\mathbf{x}^{\mathrm{A}}_{\mathrm{s}_j} - \mathbf{g}_{\mathrm{s}_j}\left(\mathbf{x}^{\mathrm{F}}, \mathbf{x}^{\mathrm{s}_j}\right)\right]\Big). \tag{37}$$

The ML estimate (the composite measurement) is then

$$\hat{\mathbf{x}}^{\mathrm{F}}_{\mathrm{ML}} = \underset{\mathbf{x}^{\mathrm{F}}}{\operatorname{argmax}}\, \mathbf{\Lambda}\left(\mathbf{x}^{\mathrm{F}}\right). \tag{38}$$

Finding the ML estimate in this case is equivalent to a nonlinear least squares problem. The numerical search will be performed via the ILS technique [2].

The ILS estimate after $(m+1)$ iterations is

$$\hat{\mathbf{x}}^{\mathrm{F}(m+1)}_{\mathrm{ILS}} = \hat{\mathbf{x}}^{\mathrm{F}(m)}_{\mathrm{ILS}} + \left[\left(\mathbf{G}^{(m)}\right)' \mathbf{R}^{-1}_{\mathrm{corr}} \mathbf{G}^{(m)}\right]^{-1} \cdot \left(\mathbf{G}^{(m)}\right)' \mathbf{R}^{-1}_{\mathrm{corr}}\left[\mathbf{z} - \mathbf{g}\left(\hat{\mathbf{x}}^{\mathrm{F}(m)}_{\mathrm{ILS}}, \mathbf{s}\right)\right], \tag{39}$$

with

$$\mathbf{z} = \left[\mathbf{x}^{\mathrm{A}}_{\mathrm{s}_1}, \mathbf{x}^{\mathrm{A}}_{\mathrm{s}_2}\right]', \tag{40}$$

$$\mathbf{s} = \left[\mathbf{x}^{\mathrm{s}_1}, \mathbf{x}^{\mathrm{s}_2}\right]', \tag{41}$$

$$\mathbf{g}\left(\hat{\mathbf{x}}^{\mathrm{F}(m)}_{\mathrm{ILS}}, \mathbf{s}\right) = [a_{\mathrm{s}_1}, e_{\mathrm{s}_1}, a_{\mathrm{s}_2}, e_{\mathrm{s}_2}], \tag{42}$$

$$\mathbf{R}_{\mathrm{corr}} = \begin{bmatrix} \mathbf{R}^{\mathrm{A}}_{\mathrm{s}_1,\mathrm{corr}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{\mathrm{A}}_{\mathrm{s}_2,\mathrm{corr}} \end{bmatrix}, \tag{43}$$

and

$$\mathbf{G}^{(m)} = \left.\frac{\partial \mathbf{g}\left(\hat{\mathbf{x}}^{\mathrm{F}(m)}_{\mathrm{ILS}}, \mathbf{s}\right)}{\partial \mathbf{x}^{\mathrm{F}}}\right|_{\mathbf{x}^{\mathrm{F}} = \hat{\mathbf{x}}^{\mathrm{F}(m)}_{\mathrm{ILS}}}, \tag{44}$$

is the Jacobian matrix of the stacked measurement vector evaluated at the ILS estimate from the previous iteration. The Jacobian matrix is given by

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{s_1} \\ \mathbf{G}_{s_2} \end{bmatrix}, \tag{45}$$

with $\mathbf{G}_{\mathrm{s}_j} = H_3$, $j = 1, 2$ as in (20).

To perform the numerical search via ILS, an initial estimate $\hat{\mathbf{x}}^{\mathrm{F}(0)}_{\mathrm{ILS}}$ is required. The initial Cartesian position is solved using the LOS measurements a_{s_1}, a_{s_2}, and a_{e_1} of the first iteration, and the known sensor positions $\mathbf{x}^{\mathrm{s}_1}$ and $\mathbf{x}^{\mathrm{s}_2}$ as follows:

$$x^{\mathrm{F}(0)}_{\mathrm{ILS}} = x^{\mathrm{s}_1} + \left(y^{\mathrm{F}(0)}_{\mathrm{ILS}} - y^{\mathrm{s}_1}\right)\tan a_{\mathrm{s}_1}, \tag{46}$$

$$y^{\mathrm{F}(0)}_{\mathrm{ILS}} = \frac{(x^{\mathrm{s}_2} - x^{\mathrm{s}_1}) + (y^{\mathrm{s}_1}\tan a_{\mathrm{s}_1} - y^{\mathrm{s}_2}\tan a_{\mathrm{s}_2})}{\tan a_{\mathrm{s}_1} - \tan a_{\mathrm{s}_2}}, \tag{47}$$

$$z^{\mathrm{F}(0)}_{\mathrm{ILS}} = z^{\mathrm{s}_1} + \tan e_{\mathrm{s}_1}\left|\frac{(x^{\mathrm{s}_1} - x^{\mathrm{s}_2})\cos a_{\mathrm{s}_2} + (y^{\mathrm{s}_1} - y^{\mathrm{s}_2})\sin a_{\mathrm{s}_2}}{\sin(a_{\mathrm{s}_1} - a_{\mathrm{s}_2})}\right|. \tag{48}$$

For the error covariance, the CRLB provides a lower bound on the estimation error obtainable from an unbiased estimator

$$\mathbb{E}\left[(\mathbf{x}^{\mathrm{F}} - \hat{\mathbf{x}}^{\mathrm{F}})(\mathbf{x}^{\mathrm{F}} - \hat{\mathbf{x}}^{\mathrm{F}})'\right] \geq \mathbf{J}^{-1}, \tag{49}$$

where $\mathbf{J}$ is the Fisher Information Matrix (FIM), $\mathbf{x}^{\mathrm{F}}$ is the true value to be estimated, and $\hat{\mathbf{x}}^{\mathrm{F}}$ is the estimate [2], [11]. The FIM is given by

$$\mathbf{J} = \mathbb{E}\left\{\left[\nabla_{\mathbf{x}^{\mathrm{F}}} \ln \mathbf{\Lambda}\left(\mathbf{x}^{\mathrm{F}}\right)\right]\left[\nabla_{\mathbf{x}^{\mathrm{F}}} \ln \mathbf{\Lambda}\left(\mathbf{x}^{\mathrm{F}}\right)\right]'\right\}\Big|_{\mathbf{x}^{\mathrm{F}} = \mathbf{x}^{\mathrm{F}}_{\mathrm{true}}}. \tag{50}$$

The gradient of the log-likelihood function is

$$\nabla_{\mathbf{x}^{\mathrm{F}}} \boldsymbol{\lambda}(\mathbf{x}^{\mathrm{F}}) = \sum_{j=1}^{2} \mathbf{G}'_{\mathrm{s}_j}\left(\mathbf{R}^{\mathrm{A}}_{\mathrm{s}_j,\mathrm{corr}}\right)^{-1}\left(\mathbf{x}^{\mathrm{A}}_{\mathrm{s}_j} - \mathbf{g}_{\mathrm{s}_j}\left(\mathbf{x}^{\mathrm{F}}, \mathbf{x}^{\mathrm{s}_j}\right)\right), \tag{51}$$

which, when plugged into, (50) gives

$$\mathbf{J} = \sum_{j=1}^{2} \mathbf{G}'_{\mathrm{s}_j}\left(\mathbf{R}^{\mathrm{A}}_{\mathrm{s}_j,\mathrm{corr}}\right)^{-1} \mathbf{G}_{\mathrm{s}_j}\Big|_{\mathbf{x}^{\mathrm{F}} = \mathbf{x}^{\mathrm{F}}_{\mathrm{true}}} = \mathbf{G}'\mathbf{R}^{-1}_{\mathrm{corr}}\mathbf{G}\Big|_{\mathbf{x}^{\mathrm{F}} = \mathbf{x}^{\mathrm{F}}_{\mathrm{true}}}. \tag{52}$$

B. Simulation Scenario

In the two-sensor setup, we have two cameras s_1 and s_2 spaced 1 km apart in global 3D cartesian space, with sensor positions $\mathbf{x}^{\mathrm{s}_1} = [x^{\mathrm{s}_1}, y^{\mathrm{s}_1}, z^{\mathrm{s}_1}]' = [-500\text{ m}, 0\text{ m}, 0\text{ m}]'$ and $\mathbf{x}^{\mathrm{s}_2} = [x^{\mathrm{s}_2}, y^{\mathrm{s}_2}, z^{\mathrm{s}_2}]' = [500\text{ m}, 0\text{ m}, 0\text{ m}]'$. Unlike the single-sensor case in Section IV-A, we consider the more realistic non-zero orientation angles of cameras 1 and 2 measuring $(\alpha_{\mathrm{s}_1}, \epsilon_{\mathrm{s}_1}, \rho_{\mathrm{s}_1}) = (24.5^\circ, 2.1^\circ, 4.5^\circ)$ and $(\alpha_{\mathrm{s}_2}, \epsilon_{\mathrm{s}_2}, \rho_{\mathrm{s}_2}) = (-2.6^\circ, -3.4^\circ, 2.8^\circ)$, respectively. It is worth noting that for every nonzero camera orientation, it will map to a reference case of our generic

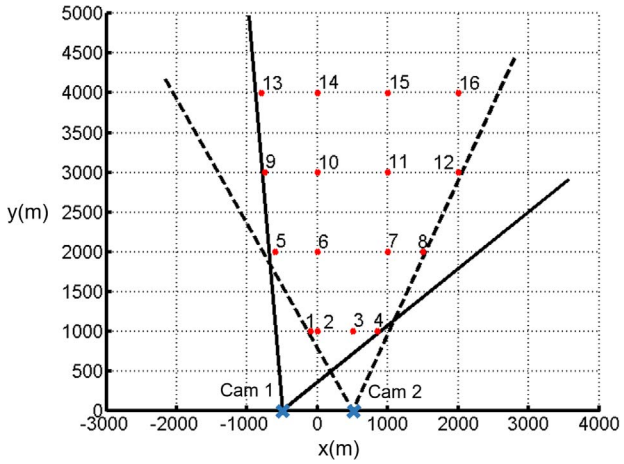


Figure 5. 2-camera setup with 16 targets in ENU space.

azimuth-elevation error ellipse characterization results we derived with camera orientation angles of zero values in Section IV-B. For this experiment, both cameras are 2-MP cameras, with the remaining camera sensor parameters and measurement error standard deviations the same as defined for the 2-MP variation of the one-sensor case in Section IV-A.

A total of 16 point targets at various ranges and heights in ENU space were considered. The 2D plot in Fig. 5 shows the camera sensor locations as crosses, with Camera 1 and 2 fields-of-view (FOV) in solid and dotted lines, respectively, and the 16 labeled targets as solid circles that are all within the FOV of both sensors. For each target point in ENU space, we plot their corresponding positions in the display image space of Camera 1 (left) and Camera 2 (right) in pixel coordinates, as shown in Fig. 6. The dotted lines segment the image space into nine general reference regions in which we have previously characterized its azimuth-elevation error ellipses in Section IV-A. Based on Fig. 6, the distribution of the targets in the image spaces of both cameras is sufficiently comprehensive to cover all scenarios for fusion.

C. Evaluation of the Composite Position Measurements and Their Error Covariances

We envisaged that the angular error covariance of our approach used in Stage 1, compared to an uncorrelated angular error covariance baseline, would translate

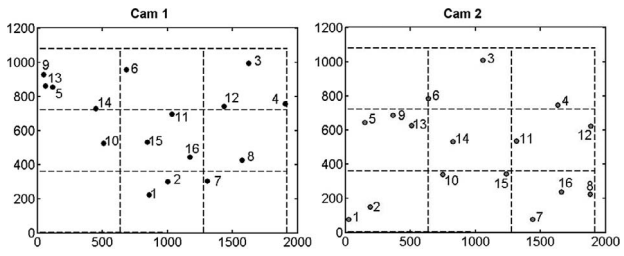


Figure 6. 16 targets in display image space of Camera 1 (left) and Camera 2 (right).

to improvements of the final composite target position measurement and its error covariance matrix. As such, we compared the composite position measurements and their corresponding Cartesian error covariances derived using

(i) the location-dependent angular error covariances, derived from the method described in Section V-A, versus
(ii) the constant uncorrelated angular error covariance baseline,

with the following evaluation criteria:

1) 3D position error improvement
2) 3D position algorithm-calculated accuracy
3) Statistical consistency of the estimates.

For each true target location in ENU space based on Fig. 5, denoted as $\check{\mathbf{x}}_i^{\mathrm{F}}, i = 1, \ldots, 16$, we get the corresponding positions in display image space of Camera 1 and Camera 2 based on Fig. 6, denoted as $\check{\mathbf{x}}_{\mathrm{s}_1,i}^{\mathrm{D}}$ and $\check{\mathbf{x}}_{\mathrm{s}_2,i}^{\mathrm{D}}, i = 1, \ldots, 16$, respectively, using the conversion

$$\check{\mathbf{x}}_{\mathrm{s}_j,i}^{\mathrm{D}} = \mathbf{h}_1^{-1}\left(\mathbf{h}_2^{-1}\left(\mathbf{T}_{\mathrm{s}_j}^{-1}\left(\alpha_{\mathrm{s}_j}, \epsilon_{\mathrm{s}_j}, \rho_{\mathrm{s}_j}\right)\left(\check{\mathbf{x}}_i^{\mathrm{F}}\right)\right)\right) \tag{53}$$

with $\mathbf{h}_1^{-1}(\cdot)$, $\mathbf{h}_2^{-1}(\cdot)$ and $\mathbf{T}_{\mathrm{s}_j}^{-1}(\cdot)$, $j = 1, 2$ the inverses of $\mathbf{h}_1(\cdot)$, $\mathbf{h}_2(\cdot)$ and $\mathbf{T}(\cdot)$ given by (1), (7), and (8), respectively. We assume the corresponding covariances in display image coordinates are given by

$$\mathbf{P}_{\mathrm{s}_j,i}^{\mathrm{D}} = \begin{bmatrix} \sigma_{P_x}^2 & 0 \\ 0 & \sigma_{P_y}^2 \end{bmatrix}, \quad j = 1, 2, \quad i = 1, \ldots, 16 \tag{54}$$

similar to what was assumed in (18).

We now derive the 3D composite target position estimates and their corresponding Cartesian error covariances of the following variations:

Case (i) - $\left(\check{\mathbf{x}}_{\mathrm{corr},i}^{\mathrm{F}}, \mathbf{P}_{\mathrm{corr},i}^{\mathrm{F}}\right)$. Following the derivation steps in Section V-A, Stage 1 is first executed by applying the conversion equations (17) and (18) with display image inputs (53) and (54) of the jth camera across $i = 1, \ldots, 16$ target points. We thus derive the camera-specific target LOS angles and the corresponding angular error covariances with location dependent variances and cross-correlated errors for each of the 16 target points, i.e., $\left(\check{\mathbf{x}}_{\mathrm{s}_j,i}^{\mathrm{A}}, \mathbf{R}_{\mathrm{s}_j,\mathrm{corr},i}^{\mathrm{A}}\right), i = 1, \ldots, 16$ and $j = 1, 2$. Stage 2 is then executed to derive the composite target position estimates (fused measurements) $\check{\mathbf{x}}_{\mathrm{corr},i}^{\mathrm{F}}$ obtained from (33) to (48) and the CRLB $\mathbf{P}_{\mathrm{corr},i}^{\mathrm{F}}$ obtained from (52) evaluated at $\mathbf{x}^{\mathrm{F}} = \check{\mathbf{x}}_{\mathrm{corr},i}^{\mathrm{F}}$ for the ith target point, $i = 1, \ldots, 16$.

The covariances are evaluated at the actual (noiseless) LOS angles (like the CRLB) and will be used to compare the uncertainties of the proposed and baseline methods.

Case (ii) - $\left(\check{\mathbf{x}}_{\mathrm{uncorr},i}^{\mathrm{F}}, \mathbf{P}_{\mathrm{uncorr},i}^{\mathrm{F}}\right)$. First, the camera-specific target LOS angles for each of the $i = 1, \ldots, 16$ target points $\check{\mathbf{x}}_{\mathrm{s}_j,i}^{\mathrm{A}}$ were derived following (17), as in

Table VI
Percentage Difference Between Volumes of the Correlated and Uncorrelated Variants of the Cartesian Measurement Error Ellipses at Each of 16 Target Points $\breve{\mathbf{x}}_i^{\mathrm{F}}$, $i = 1, \ldots, 16$

Tgt	1	2	3	4	5	6	7	8
Vol diff. (%)	−8.9	1.7	8.0	**−19.9**	**−25.0**	**18.2**	7.7	**−21.3**
Tgt	9	10	11	12	13	14	15	16
Vol diff. (%)	**−20.4**	**19.2**	**25.9**	**−13.9**	**−13.5**	**17.6**	**27.2**	3.7

Case (i). The baseline uncorrelated angular error covariances $\mathbf{R}^{\mathrm{A}}_{\mathrm{s}_j,\mathrm{uncorr},i}$, $i = 1, \ldots, 16$ and $j = 1, 2$ follow according to (16) with[5] $\sigma_a = \sigma_e = \sigma_{P_x} \frac{\Theta_x}{P_x}$. Similarly, the composite target position estimates $\breve{\mathbf{x}}^{\mathrm{F}}_{\mathrm{uncorr},i}$ are obtained from (33) to (48) and the CRLB $\mathbf{P}^{\mathrm{F}}_{\mathrm{uncorr},i}$ is obtained from (52) evaluated at $\mathbf{x}^{\mathrm{F}} = \breve{\mathbf{x}}^{\mathrm{F}}_{\mathrm{uncorr},i}$ for the ith target point, $i = 1, \ldots, 16$, with the correlated covariances $\mathbf{R}^{\mathrm{A}}_{\mathrm{s}_j,\mathrm{corr},i}$ in the equations replaced with their uncorrelated counterpart $\mathbf{R}^{\mathrm{A}}_{\mathrm{s}_j,\mathrm{uncorr},i}$.

We now compare and evaluate our proposed approach according to each of the three criteria listed above.

1) 3D position error improvement: The percentage difference between the volumes of the correlated and uncorrelated variants of the Cartesian measurement error ellipses is calculated by

$$\mathrm{d} = \frac{\pi\left(\sqrt{\det\left(\mathbf{P}^{\mathrm{F}}_{\mathrm{corr}}\right)} - \sqrt{\det\left(\mathbf{P}^{\mathrm{F}}_{\mathrm{uncorr}}\right)}\right)}{\pi\sqrt{\det\left(\mathbf{P}^{\mathrm{F}}_{\mathrm{uncorr}}\right)}} \times 100\%, \quad (55)$$

where $\det(\mathbf{A})$ is the determinant of matrix $\mathbf{A}$, and the results are tabulated in Table VI for each of the 16 selected target points. A magnitude of $|\mathrm{d}| \geq 10\%$ will imply that the difference between the correlated and uncorrelated variants of the Cartesian measurement error ellipses is significant, and these values are highlighted in bold in Table VI.

It is observed that the majority of target points have significant differences in volume between the correlated and uncorrelated variants of the Cartesian measurement error ellipses, which demonstrates the importance of using our proposed conversion method for improved error modeling. Furthermore, our approach can yield up to a maximum of approximately 27% difference in volume of the correlated Cartesian measurement error ellipse relative to the baseline uncorrelated Cartesian measurement error ellipse.

2) 3D position accuracy improvement: At this point, we generate a set of $N = 1000$ random points with Gaussian noise in the display image space of Cameras 1 and 2 for each selected target point, which models the probability distributions in the display image space

$$\left\{\mathbf{x}^{\mathrm{D}(k)}_{\mathrm{s}_j,i}\right\}_{k=1}^{N} \sim \mathcal{N}\left(\breve{\mathbf{x}}^{\mathrm{D}}_{\mathrm{s}_j,i}, \mathbf{R}^{\mathrm{D}}_{\mathrm{s}_j,i}\right) \quad j = 1, 2, \quad i = 1, \ldots, 16. \quad (56)$$

Each set of points undergoes the non-linear transformation $\mathbf{h}(\cdot)$ defined in (17) to derive a distribution of points

$$\left\{\mathbf{x}^{\mathrm{A}(k)}_{\mathrm{s}_j,i}\right\}_{k=1}^{N} \quad j = 1, 2, \quad i = 1, \ldots, 16, \quad (57)$$

that model the probability distribution in azimuth-elevation space corresponding to the ENU coordinate system. For *Case (i)*, the corresponding set of correlated angular error covariances

$$\left\{\mathbf{R}^{\mathrm{A}(k)}_{\mathrm{s}_j,\mathrm{corr},i}\right\}_{k=1}^{N} \quad j = 1, 2, \quad i = 1, \ldots, 16, \quad (58)$$

are derived from the transformation (18) based on the random angular state points specified by (57). For *Case (ii)*, the corresponding set of baseline uncorrelated angular error covariances

$$\left\{\mathbf{R}^{\mathrm{A}(k)}_{\mathrm{s}_j,\mathrm{uncorr},i}\right\}_{k=1}^{N} \quad j = 1, 2, \quad i = 1, \ldots, 16, \quad (59)$$

assume fixed values as specified by (16).

Subsequently, we apply equations (33)–(48) and (52) to derive the composite target estimates

$$\left\{\hat{\mathbf{x}}^{\mathrm{F}(k)}_{\mathrm{corr},i}, \mathbf{P}^{\mathrm{F}(k)}_{\mathrm{corr},i}\right\}_{k=1}^{N} \quad i = 1, \ldots, 16, \quad (60)$$

for *Case (i)*. The kth composite position estimate of the i^{th} target $\hat{\mathbf{x}}^{\mathrm{F}(k)}_{\mathrm{corr},i}$ is based on the fusion of the kth random angular points, and their respective covariances from camera sensors s_1 and s_2, $\left(\mathbf{x}^{\mathrm{A}(k)}_{\mathrm{s}_1,i}, \mathbf{R}^{\mathrm{A}(k)}_{\mathrm{s}_1,\mathrm{corr},i}\right)$ and $\left(\mathbf{x}^{\mathrm{A}(k)}_{\mathrm{s}_2,i}, \mathbf{R}^{\mathrm{A}(k)}_{\mathrm{s}_2,\mathrm{corr},i}\right)$, specified in (57) and (58). The corresponding kth Cartesian error covariance estimate of the ith target $\mathbf{P}^{\mathrm{F}(k)}_{\mathrm{corr},i}$ follows from (52) evaluated at $\mathbf{x}^{\mathrm{F}} = \hat{\mathbf{x}}^{\mathrm{F}(k)}_{\mathrm{corr},i}$.

In a similar manner, we apply equations (33)–(48) and (52) to derive the composite target estimates and their covariances

$$\left\{\hat{\mathbf{x}}^{\mathrm{F}(k)}_{\mathrm{uncorr},i}, \mathbf{P}^{\mathrm{F}(k)}_{\mathrm{uncorr},i}\right\}_{k=1}^{N} \quad i = 1, \ldots, 16 \quad (61)$$

for *Case (ii)*, based on the random angular point sets specified by (57) and (59), and with (52) evaluated at $\mathbf{x}^{\mathrm{F}} = \hat{\mathbf{x}}^{\mathrm{F}(k)}_{\mathrm{uncorr},i}$.

We calculate the differences of the true composite target state with the correlated variant of the composite position estimate

$$\Delta\mathbf{x}^{\mathrm{F}(k)}_{\mathrm{corr},i} = \left(\hat{\mathbf{x}}^{\mathrm{F}(k)}_{\mathrm{corr},i} - \breve{\mathbf{x}}^{\mathrm{F}}_i\right)' \left(\hat{\mathbf{x}}^{\mathrm{F}(k)}_{\mathrm{corr},i} - \breve{\mathbf{x}}^{\mathrm{F}}_i\right) \quad i = 1, \ldots, 16, \quad (62)$$

and the uncorrelated variant of the composite position estimate

$$\Delta\mathbf{x}^{\mathrm{F}(k)}_{\mathrm{uncorr},i} = \left(\hat{\mathbf{x}}^{\mathrm{F}(k)}_{\mathrm{uncorr},i} - \breve{\mathbf{x}}^{\mathrm{F}}_i\right)' \left(\hat{\mathbf{x}}^{\mathrm{F}(k)}_{\mathrm{uncorr},i} - \breve{\mathbf{x}}^{\mathrm{F}}_i\right) \quad i = 1, \ldots, 16, \quad (63)$$

[5]These are the "simplistic" constant measurement accuracies.

where $\check{\mathbf{x}}_i^{\mathrm{F}}$ are the true target locations in ENU space.

Following the above, both correlated and uncorrelated variants of the overall root mean square error (RMSE) across $N = 1000$ sample runs are calculated in (64) and (65), respectively

$$\mathrm{RMSE}_{\mathrm{corr},i} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} \Delta\mathbf{x}_{\mathrm{corr},i}^{\mathrm{F}(k)}} \qquad i = 1, \ldots, 16, \quad (64)$$

$$\mathrm{RMSE}_{\mathrm{uncorr},i} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} \Delta\mathbf{x}_{\mathrm{uncorr},i}^{\mathrm{F}(k)}} \qquad i = 1, \ldots, 16. \quad (65)$$

Additionally, the percentage difference between the overall RMSE values of the correlated and uncorrelated variants of the 3D composite position estimates is calculated by

$$\mathrm{RMSE\_diff} = \frac{\mathrm{RMSE}_{\mathrm{corr}} - \mathrm{RMSE}_{\mathrm{uncorr}}}{\mathrm{RMSE}_{\mathrm{uncorr}}} \times 100\%, \quad (66)$$

and the results are tabulated in Table VII for each of the 16 selected target points.

It is observed that our approach using the more appropriate correlated angular error covariance matrices yields up to a maximum of approximately 0.7% improvement in 3D positional accuracy relative to the composite position estimate derived from the baseline uncorrelated angular error ellipses. Across all targets, the 3D positional accuracy differences are insignificant. This is because in both correlated and uncorrelated cases, we used the same transformation in (17) to convert measurements from the display image space to the azimuth-elevation space. The azimuth and elevation values for each point set were thus identical in both cases. As a result, the triangulation estimates of the two methods are similar. Although the measurement error covariances in (58) and (59) differ, their contributions do not significantly impact the RMSE. While Table VII showed little differences, Table VI showed significant differences for the algorithm-calculated covariances (which are used in recursive state estimation).

3) Statistical Consistency: To test the statistical consistency of the correlated variant of the composite position estimate, the normalized estimation error squared (NEES) [2] is used, with the CRLB $\mathbf{P}_{\mathrm{corr},i}^{\mathrm{F}}$ as the covariance matrix, i.e., the inverse of the FIM in (52) evaluated at $\mathbf{x}^{\mathrm{F}} = \check{\mathbf{x}}_{\mathrm{corr},i}^{\mathrm{F}}$ for the ith target point, $i = 1, \ldots, 16$. Along with definition (62), we have that

$$\epsilon_{\mathrm{corr},i}^{\mathrm{F}(k)} = \left(\Delta\mathbf{x}_{\mathrm{corr},i}^{\mathrm{F}(k)}\right)' \mathbf{P}_{\mathrm{corr},i}^{\mathrm{F}\ -1} \left(\Delta\mathbf{x}_{\mathrm{corr},i}^{\mathrm{F}(k)}\right) \sim \chi_{n_x}^2, \quad (67)$$

and thus the chi-squared test statistic

$$\bar{\epsilon}_{\mathrm{corr},i}^{\mathrm{F}} = \frac{1}{N}\sum_{k=1}^{N} \epsilon_{\mathrm{corr},i}^{\mathrm{F}(k)} \sim \frac{1}{N}\chi_{Nn_x}^2, \quad (68)$$

With $N = 1000, n_x = 3$ and $\alpha = 0.95$, the two-tailed 95% interval is given by

$$\bar{\epsilon}_{\mathrm{corr},i}^{\mathrm{F}} \in [r_1, r_2] \quad i = 1, \ldots, 16, \quad (69)$$

where $[r_1, r_2] = \left[\frac{1}{1000}\chi_{3000}^2(0.025), \frac{1}{1000}\chi_{3000}^2(0.975)\right] = [2.8501, 3.1537]$.

Similarly, we repeat steps (67)–(69) to test the statistical consistency of the baseline uncorrelated variant of the composite position estimate. We still evaluate this uncorrelated variant of the estimate at the same CRLB $\mathbf{P}_{\mathrm{corr},i}^{\mathrm{F}}$, and replace definition (62) with (63) to derive the uncorrelated versions $\epsilon_{\mathrm{uncorr},i}^{\mathrm{F}(k)}$ and $\bar{\epsilon}_{\mathrm{uncorr},i}^{\mathrm{F}}$. Note that $\bar{\epsilon}_{\mathrm{uncorr},i}^{\mathrm{F}}$ shares the same two-tailed 95% test interval as $\bar{\epsilon}_{\mathrm{corr},i}^{\mathrm{F}}$.

Figure 7 shows the plot of the sample average NEES from $N = 1000$ runs of the correlated (red-filled circles) and uncorrelated (black-filled triangles) variants of the composite position estimate, using the same appropriate CRLB as the covariance matrix, for the 16 selected targets. The dotted lines within the plot indicate the 95% two-tailed interval calculated as per (69). Since all chi-squared test statistics across all 16 targets fall within the 95% two-tailed interval, the composite position estimates derived from both our correlated and the uncorrelated baseline approaches are statistically consistent.

Thus, the benefit of the proposed method of fusion is in the correctly calculated uncertainty region volumes presented in Table VI, even though the estimates of the

Table VII
Absolute RMSE Values and Percentage Difference between RMSE Values of the Correlated and Uncorrelated Variants of the Composite 3D Cartesian Position Estimates Evaluated at Each of 16 True Target Points $\check{\mathbf{x}}_i^{\mathrm{F}}, i = 1, \ldots, 16$

Tgt	1	2	3	4	5	6	7	8
RMSE [corr] (m)	1.127	1.191	1.463	1.745	3.534	3.650	4.957	5.945
RMSE [uncorr] (m)	1.131	1.192	1.466	1.758	3.536	3.656	4.968	5.958
RMSE diff. (%)	0.332	0.134	0.209	0.714	0.045	0.179	0.221	0.219
Tgt	9	10	11	12	13	14	15	16
RMSE [corr] (m)	7.495	7.814	9.210	12.136	13.017	13.731	15.285	18.008
RMSE [uncorr] (m)	7.484	7.820	9.209	12.146	13.025	13.720	15.280	18.019
RMSE diff. (%)	− 0.150	0.065	− 0.005	0.082	0.058	− 0.076	− 0.031	0.062

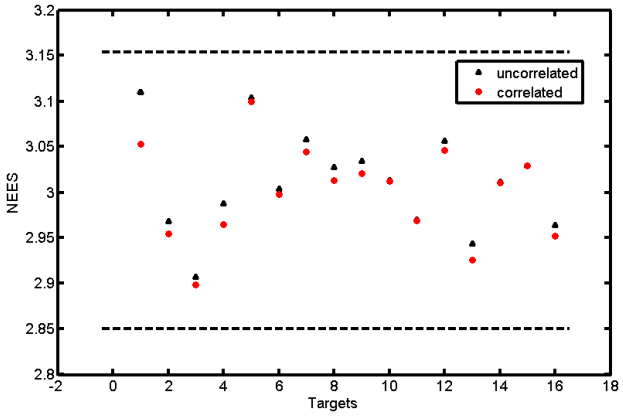


Figure 7. Sample average NEES for correlated and uncorrelated variants of the composite position estimate for targets $i = 1, \ldots, 16$.

proposed and baseline approaches are not significantly different.

VI. CONCLUSIONS

In this paper, we devised an approach to convert the raw target measurements and their covariance in the image space of EO cameras to the azimuth and elevation angle measurements in the global ENU coordinate system, which is validated to be unbiased, with FPA-location-dependent variances and cross-correlated errors. Typical tracking applications usually assume that azimuth and elevation measurement errors have the same standard deviation are uncorrelated and uniform across the camera's FOV, which is not the case. Simulations were performed for nine selected points on the FPA coming from both 2MP and 8MP commercially available camera sensors, with comparisons of the bias error and correlated covariances to an uncorrelated baseline error measure. Results show that cross-correlated errors are present and significant enough to be considered. The magnitudes of variances and correlations are dependent on the target's location with respect to the camera center. In addition, the angular estimate derived from the conversion has been shown to be unbiased, and the derived covariance matrix with cross-correlated errors has been shown to be consistent, based on the statistical tests conducted.

With these findings, we envisioned an unbiased converted angular measurement and a more appropriate measurement error covariance accounting for cross-correlated errors should be used as input to the 3D Cartesian state filtering equations and multisensor fusion for improved tracking performance. To demonstrate this, we considered a multisensor fusion application to derive a composite target position measurement and its corresponding Cartesian measurement error covariance in 3D Cartesian space. This was done for a two-camera setup, using LOS target angle measurements with the more appropriate measurement error covariances of our method. Simulations were performed for 16 selected target points in ENU space. Results show that the composite 3D target position and its corresponding Cartesian measurement error covariance estimates derived using the LOS angular measurements and angular measurement error covariance of our approach are statistically efficient. Compared to the uncorrelated LOS angular error covariances, our approach in accounting for location-dependent variances and cross-correlated errors in the LOS angular error covariances translated to a significant improvement to the error modeling of the Cartesian measurement error ellipse, but minimal improvement to the 3D Cartesian position accuracy of the target. Since in filtering the gain depends on the measurement covariance, the proposed approach will yield superior performance. The impact of using the correlated covariance for the measurements on the filtering performance is a topic of future research. It is envisaged that the correct error covariance is crucial in real-world applications, particularly for tracking in cluttered environments or closely spaced targets. Accurate measurement error covariance enhances track-to-measurement association accuracy. The additional computational cost, which is not significant, is well worth it.

REFERENCES

[1] B. Balaji, R. Sithiravel, Z. Daya, and T. Kirubarajan "Aspects of detection and tracking of ground targets from an airborne EO/IR sensor," *Proc. SPIE*, vol. 9474, Art. no. 947403.

[2] 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.

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

[4] X. Liu, W. Chen, H. Madhusudanan, L. Du, and Y. Sun "Camera orientation optimization in stereo vision systems for low measurement error," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 2, pp. 1178–1182, Apr. 2021.

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

[6] M. Mallick "Geolocation using video sensor measurements," *Proc. 10th Int. Conf. Inf. Fusion*, Quebec, QC, Canada, 2007, pp. 1–8.

[7] M. Mallick, X. Tian, Y. Zhu, and M. Morelande "Angle-only filtering of a maneuvering target in 3D," *Sensors*, vol. 22, no. 4, Art. no. 1422, 2022.

[8] L. Mejias, J. J. Ford, and J. Lai "Towards the implementation of vision-based UAS sense-and-avoid system," Computational Statistics & Data Analysis, *Proc. 27th Int. Congr. Aeronautical Sci., Acropolis Conference Centre*, Nice, France, Sep. 2010.

[9] A. Miller and B. Miller "Tracking of the UAV trajectory on the basis of bearing-only observations,"

in *Proc. 53rd IEEE Conf. Decis. Control*, Los Angeles, CA, USA, 2014, pp. 4178–4184.

[10] S. Omar and S. Winberg
"Multisensor data fusion: Target tracking with a Doppler radar and an electro-optic camera,"
Proc. 2011 IEEE Int. Conf. Control System, Computing and Engineering, Penang, Malaysia, 2011, pp. 210–215.

[11] 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.

[12] R. Yang and Y. Bar-Shalom
"Full state information transfer across adjacent cameras in a network using gauss helmert filters,"
J. Adv. Inf. Fusion, 17(1), pp. 14–28, Jun. 2022.

[13] R. Yang, Y. Bar-Shalom, and H. A. J. Huang
"Camera calibration using inaccurate and asynchronous discrete GPS trajectory from drones,"
J. Adv. Inf. Fusion, vol. 18, no. 1, pp. 3–15, Jun. 2023.

Jessica Koon Yan Goh received the B.Sc. and Master of Mathematics and Statistics degree from the University of Warwick, Coventry, UK, in 2017. She is currently a Senior Member of Technical Staff at DSO National Laboratories, Singapore. Her research interests include target tracking and multisensor data fusion.

Yaakov Bar-Shalom (F'84) received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion in 1963, Haifa, Israel 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 Dept. 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 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 Transactions on Automatic Control and Automatica. He was General Chairman of the 1985 ACC, General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications from 2004 to 2013. Since 1995, he has been a Distinguished Lecturer of the IEEE AESS. He is a corecipient of the M. Barry Carlton Award for the best paper in the IEEE TAE Systems 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 2022, the IEEE AESS Pioneer Award. He has been listed by academic.research.microsoft (top authors in engineering) as #1 among the researchers in aerospace engineering based on the citations of his work. 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.

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

Sliding Window Estimation Based on PEM for Visual/Inertial SLAM

ZORAN SJANIC
MARTIN A. SKOGLUND

This paper presents a sliding window estimation method for simultaneous localization and mapping (SLAM) based on the prediction error method (PEM). The estimation problem considers landmarks as parameters while treating dynamics using state space models. The gradient needed for parameter estimation is computed recursively using an extended kalman filter. Results from experiments and simulations with a monocular camera and inertial sensors are presented and compared to batch PEM and nonlinear least-squares SLAM estimators. The presented method maintains good accuracy, and its parametrization is well-suited for online implementation, as it scales better with the size of the problem than batch methods.

Manuscript received October 4, 2023; revised June 11, 2024; released for publication October 14, 2024

Associate Editor handling the review was Ramona Georgescu. The authors gratefully acknowledge funding from the Vinnova Industry Excellence Center LINK-SIC.

Z. Sjanic is with the Department of Automatic Control, 581 83 Linköping University, Linköping, Sweden, and also with the Saab AB, Linköping, Sweden (e-mail: Zoran.Sjanic@liu.se).

M. A. Skoglund is with the Department of Automatic Control, 581 83 Linköping University, Linköping, Sweden, and also with the Eriksholm Research Centre, Oticon A/S, 3070 Snekkersten, Denmark (e-mail: Martin.Skoglund@liu.se).

I. INTRODUCTION

The work in this paper introduces the use of the prediction error method (PEM), see, e.g., [18], as well as its extension to sliding window (SW) as a way to utilize the particular structure of problems encountered in simultaneous localization and mapping (SLAM). The aim in SLAM is to estimate a moving platform's position and orientation while simultaneously mapping the observed environment [1], [7]. A strong trend in SLAM algorithm research is (incremental) batch optimization, which usually solves some form of nonlinear least squares (NLS) problem, see for example [4], [12], [20], [23], [24], [28], [31], [32], [34], [35]. Due to the problem's nature, where both the platform's motion and the environment are considered unknown parameters, solvers are computationally expensive, typically quadratic in the length of the motion parameters; see, e.g., [27]. Throughout the years, many methods were devised aiming at reducing this computational cost. Some of them are utilizing the sparsity of the involved matrices during the calculation in order to use sparse solvers, see, e.g., [10], [11], [20], while others aim at dimensionality reduction of the original problem by solving for only motion parameters while the map is implicit [14], [32] (leading to GraphSLAM), or vice versa by submaps [35]. One issue with the sparsity of the SLAM problem is that it will vary with the problem instance, i.e., the actual environment and the motion. Other methods are utilizing the particular structure of the SLAM problem and aiming at decoupling of the mapping and localization parts, e.g., [26], [27], [29], [33]. These kinds of methods utilizes the intrinsic property of the problem independently of the actual problem instance. Note also that the decoupling must be done in such a way that the correlation between landmarks and the motion is preserved, as it is an inherent property of the SLAM posterior distributions.

The main idea adopted in this work is to model the whole system, i.e., the moving platform and the environment, as a parameterized dynamic model. The environment is represented with discrete points (also called landmarks), which are considered to be the parameters of the system, while the motion of the platform is modeled as a dynamic system. The motivation behind this formulation is that the solution to the problem can be split into two parts, the first part where the landmarks are estimated, and the second part where the motion is estimated. The motion states estimate is here used as a predictor of the system output while utilizing the time series properties through filter solutions. Worth noting is that this property also allows for the gradient of the predictor w.r.t. parameters to be calculated recursively (for a particular choice of the predictor) so that no numerical gradients are necessary. This approach differs from both the standard extended Kalman filter (EKF)-SLAM approach, see, e.g., [7], where both the platform's motion and the landmarks are considered as dynamic states, and the NLS approaches, where everything is considered

as (static) parameters. This division into two parts leads to computation complexity reduction in the following sense:

- The landmark estimation is an optimization problem that is smaller (in the number of estimated parameters) than the standard NLS problem.
- The motion estimation becomes the prediction problem, which is smaller than the usual EKF-SLAM problem since the number of states is constant, i.e., only the motion states are used.

In summary, the optimization problem will scale with the number of parameters (landmarks), while the predictor part will scale with the measurement batch length (time steps). Unlike the approach in [33], the solution proposed here circumvents the need to transform the absolute measurements, i.e., from the platform to the landmarks, to the relative measurements, i.e., between the landmarks. This transformation assumes that range and bearing to the landmarks are measured, and would not work with the original (visual/bearing only) measurements.

The PEM-SLAM batch approach was described in [26], and the extension is to consider an SW adaptation that might enable an efficient online implementation, see e.g., [6], [25]. Online versions of PEM are quite unusual, see, for example, [19], [30], perhaps as excitation and convergence are much harder to obtain and prove. Since the window is just a shorter batch, the SW-PEM-SLAM will consequently be more computationally efficient than SW-NLS-SLAM (given the same window length). This will be shown in an empirical way with Monte Carlo (MC) simulations. The adaptation of PEM-SLAM to an SW is the main contribution of this paper. This adaption is primarily aiming for potential online applications of PEM-SLAM as an estimation method.

The outline of the paper is as follows: In Section II, a brief description of PEM and the model structure is given; In Section III, the expansion of the PEM to the Sliding Window approach is described; In Section IV, the computation complexity of the suggested approach is analyzed; In Section V, the example dynamic and measurements models that are used for evaluation are presented. In Section VI, MC simulation results as well as real data experiments are presented and compared to (SW-)NLS-SLAM, and some empirical computation complexity comparison to (SW-)NLS-SLAM; Section VII ends the paper with conclusions and directions for future work.

II. THE PREDICTION ERROR METHOD

Assume that measurements, $\{y_t\}_{t=1}^N$, from a dynamic system are available. Suppose also that a model of this system is parametrised with some unknown parameters, $\Theta = \{\theta^l\}_{l=1}^L$, and that we want to estimate these using PEM. For that purpose we can use a (quite general) system description of a discrete-time nonlinear state space model as

$$x_{t+1} = f_t(x_t, u_t, w_t, \Theta), \tag{1a}$$

$$y_t = h_t(x_t, \Theta) + e_t. \tag{1b}$$

In this system, the state dynamics is modeled with the function $f_t(\cdot)$, u_t is a known input, w_t is an unknown system noise, the measurement-to-state relation is represented with function $h_t(\cdot)$ and e_t is the measurement noise. The one-step ahead measurement predictor, i.e., predicted measurements at time t given all the information until time $t-1$, is obtained by plugging the predicted state $\hat{x}_{t|t-1}$ into (1b) giving $\hat{y}_{t|t-1}(\Theta) = h_t(\hat{x}_{t|t-1}, \Theta)$. The predictor can, for example, be implemented as an EKF. PEM estimation of Θ is then done by minimizing the sum of the squared norms of the prediction errors, defined as

$$\hat{\Theta} = \arg\min_{\Theta} V(\Theta), \tag{2}$$

and where we have defined $V(\Theta)$ as

$$V(\Theta) = \frac{1}{N}\sum_{t=1}^{N} \mathcal{L}(y_t - \hat{y}_{t|t-1}(\Theta)). \tag{3}$$

$\mathcal{L}$ can be any positive function and $\hat{y}_{t|t-1}(\Theta)$ is defined above. Usually, $\mathcal{L}$ is a standard 2-norm, i.e., $\mathcal{L}(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, leading to the standard (possibly nonlinear) least-squares method. Note that robust norms, such as Huber, can account for spurious data association, see, e.g., [4], but these cases are beyond the scope of this paper.

In this work, we adopt the 2-norm cost function and (3) becomes a standard NLS problem, and any NLS method, such as the Levenberg–Marquardt method [16], [21], can be used to solve it. In this context, it is quite advantageous if the predictor is an analytical function of the parameters and if its gradient w.r.t. to parameters is available. These both qualities simplify and speed up the iterative optimization procedure compared to using numerical methods for calculating the gradient.

A. System Properties and Choice of Predictor

PEM formulations, such as (3), typically result in predictors, $\hat{y}_{t|t-1}(\Theta)$, that are nonlinear in the parameters even for linear Gaussian state space models in (1a). In [17], a recursive PEM method based on the EKF predictor where parameters are appended to the state vector is analyzed. It is shown to be globally asymptotically convergent for the general case (and for linear dynamic systems). This motivates our use of EKF as a predictor in a similar manner to, e.g., [13], [15]. Another advantage is its simple implementation and its possibility to explicitly calculate the gradient of the loss function in (3).

Furthermore, the predictor, $\hat{y}_{t|t-1}(\Theta)$, that will be used (i.e., pinhole camera projection), see Section V-C, is a convex function in parameters, see, e.g., [2], which is a good property for optimization. Another good PEM

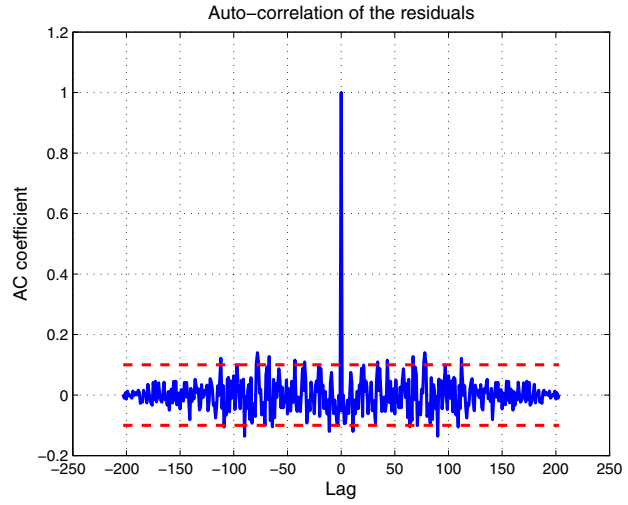


Figure 1. Auto-correlations for the residuals for one landmark and one dimension over the whole trajectory averaged over 50 MC runs. A total of 90% limit is also shown with the red dashed line.

property is that, if the true system is in the model set, if the predictor is stable, and if the innovations are white, then the parameters will converge to true ones; see [18]. In the case for PEM-SLAM here, the model set is modeling the landmarks, which are observed, and consequently it is covering the true environment. The innovations from EKF are (approximately) white, see Fig. 1 where simulation data explained in Section VI-A are used. The residuals' auto-correlations, averaged over 50 MC simulations, are shown for one landmark and in one dimension as an example, but the behavior is very similar for the rest of the landmarks. This fact further motivates the choice of EKF as a predictor. Note that residuals may have heavy-tailed distributions before convergence due to the initialization procedure and also if outliers are not removed.

All the above-mentioned facts support the usage of PEM as a method for solving the SLAM problem. However, with recursive PEM using EKF, the parameters are included in the state vector (with zero dynamics). This is similar to EKF-SLAM, where landmarks are considered as states, [7]. This resemblance highlights the drawbacks of straightforwardly applying PEM to standard EKF-SLAM, which would be severely limited to small maps. Instead, we devise another kind of PEM adaptation, namely SW PEM, which is a main contribution of this paper, and which will be explained in the next section.

III. SLIDING WINDOW PEM ESTIMATION

An adaptation of the PEM is done to accommodate real-time estimation, in a fashion similar to a filter. The adaptation is an iterative procedure where the problem defined in (3) is solved iteratively inside an SW of length K where $K \leq N$. The resulting cost function used in each iteration, i, is then (i is increased by one after each iteration)

$$V_i(\Theta_i) = \frac{1}{2K} \sum_{t=i}^{i+K} \|y_t - \hat{y}_{t|t-1}(\Theta_i)\|_2^2, \tag{4a}$$

$$\hat{\Theta}_i = \arg\min_{\Theta_i} V_i(\Theta_i). \tag{4b}$$

Caution must be taken here when it comes to parameter vector Θ since not all of the parameters might be observed in the window. Hence, Θ_i denotes the subset of the parameters that are observed in the iteration window i. Then, a locally weak observability condition is simply that there are at least as many measurements as parameters in each window and that the resulting observability rank condition [9] is satisfied. The predictor (EKF) in each iteration is initiated by the state and covariance estimate obtained from the previous iteration for the time instance $t = i + 1$ (since this is the first value of the state and covariance in iteration $i + 1$).

A. Statistical Approximation for Information Fusion

To obtain a single best estimate of the static parameters in Θ, we need to fuse different ones from each window. However, double counting of information is a potential problem that can lead to overly confident estimates. This needs to be addressed when parameters are estimated in the moving window. For example, it is very likely that a certain parameter $\theta^l \in \Theta_i$ will be observed and estimated in the window i and $i+1$. In order to handle this (at least approximately), two things are applied:

- The current best parameter estimate is saved.
- The parameter estimates are fused from from the non-overlapping windows only.

The motivation for keeping the current best estimate is that we want to use these in the fusion step, since this will add most information. Also, the estimates from the non-overlapping windows are based on different measurements, which would make them statistically independent. However, this approach is only an approximation since the estimates are dependent on the predictor which, in turn, is dependent on all the past measurements. This dependency will become weaker the further apart in time the state estimates are. Therefore, an assumption is that the estimates from different windows have limited dependency. This is deemed acceptable for our purposes. In the explanation of the fusion principle below, we will omit superindex l in θ^l as notation for an arbitrary parameter in the parameter vector for enhanced readability purposes.

The currently best estimate of the parameter, $\hat{\theta}_b$, is calculated based on an information criterion using the Jacobian $J(\hat{\theta}) = \frac{\partial V(\Theta)}{\partial \theta}\big|_{\theta=\hat{\theta}}$ evaluated with $\hat{\theta}$ from (4b).

Algorithm 1 Sliding Window PEM.

Require: $\hat{x}_{1|1}, P_{1|1}, y_{1:N}, K$
Ensure: $\hat{\Theta}, \hat{x}_{1:N}, P_{1:N}$

for $i = 1 : K - N$
1. Set initial state and covariance for the predictor to $\hat{x}_{i|i}$ and $P_{i|i}$
2. Solve the minimization problem with $V_i(\Theta_i)$ as in (4a)
if $\hat{\theta}^l$ is new parameter **then**
3. Set $\hat{\theta}^l_b$ and $\hat{\theta}^l_f$ to that estimate
else
4. Save the best estimate of the parameters, $\hat{\Theta}_b$, according to (5)
end if
if $\hat{\theta}^l_b$ not updated **and** $\hat{\theta}^l_f$ and $\hat{\theta}^l_b$ are estimated in nonoverlapping windows **then**
5. Fuse $\hat{\theta}^l_f$ and $\hat{\theta}^l_b$ according to (6)
end if
6. Save $\hat{x}_{i|i}$ and $P_{i|i}$
end for
7. $\hat{\Theta} \coloneqq \hat{\Theta}_f$

The following procedure is used:

$$\hat{\theta}_b = \begin{cases} \hat{\theta}_i & \text{if } \mathrm{Tr}(\mathcal{I}_\mathrm{i}) < \mathrm{Tr}(\mathcal{I}_\mathrm{b}). \\ \hat{\theta}_b & \text{otherwise} \end{cases} \tag{5}$$

where index b denominates the window with the best estimate, index i is the current window, and $\mathcal{I} = J(\hat{\theta})^T J(\hat{\theta})$ is the information matrix for the parameter. In this way, we assure that any new estimate will not override the currently best one. We also store the information matrix, $\mathcal{I}_b$ and the window (start and end time indices) together with the current best estimate.

Since we want to utilize all the available information, and get the overall best estimate, fusion from non-overlapping windows is performed. This is done by keeping the fused estimate and the last time index when the fusion was performed. If the parameter's best estimate according to (5) is not updated in the last window, a check is performed to see if the best estimate comes from a window that is not overlapping with the current fused estimate, $\hat{\theta}_f$. In that case, they are fused according to their information representation

$$\hat{\theta}_f = (\mathcal{I}_f + \mathcal{I}_b)^{-1}(\mathcal{I}_f\hat{\theta}_f + \mathcal{I}_b\hat{\theta}_b), \tag{6a}$$

$$\mathcal{I}_f = \mathcal{I}_f + \mathcal{I}_b. \tag{6b}$$

The time index of the fused parameter is then updated accordingly. It shall also be pointed out that the requirement on the nonoverlapping windows is necessary to guarantee that the fused estimates are independent, i.e., estimated with different measurements. The whole approach is summarized as pseudo-code in Algorithm 1.

IV. COMPUTATION COMPLEXITY ANALYSIS

In this section, qualitative computation complexity based on the order of magnitude ($\mathcal{O}$-notation) of the suggested method will be analyzed and compared to primarily NLS method. NLS is chosen since it is *de facto* the standard method and all our numerical results are compared to it. Also, just as in analysis presented in [27], the GraphSLAM method will briefly be mentioned, but no numerical comparison is done. Since this analysis is very similar to the one in the reference above, many details will therefore be omitted here, and the reader is referred to the reference for more information.

The following assumptions and notation will be used: Window size will be denoted by K, and the number of measurements at time t is N_t, and consequently the total number of measurements in the window i is $N_i = \sum_{t=i}^{i+K} \mathsf{N}_t$ (which is linear in the number of time steps in the window, K). The total number of the observed landmarks in the window i is denoted by M_i. Define further the total number of windows as $N_w = N - K + 1$ and the average number of measurements per window as $\bar{N} = \sum_i N_i/N_w$ and the average number of landmarks per window as $\bar{M}$ in the same manner. The average number of landmark measurements per time step is then $\bar{N}^K = \bar{N}/K$, and the average number of measurements per landmark (in a given window) is $\bar{N}^M = \bar{N}/\bar{M}$. We will also assume that the main complexity lies in the calculation of the Jacobian during the iterative optimization procedure, and the analysis will be concentrated on that kind of calculation, i.e., the computation complexity for one iteration step. Notice also that no consideration is taken to any possible sparse structure of the Jacobian and its influence on the solution speed, because that is very much data dependent and will vary between the problem instances.

A. SW-PEM-SLAM

For the SW-PEM-SLAM, the Jacobian of the loss function in (4a) contains only the partial derivatives of the residuals with respect to the landmarks. Since there are, in average, $\bar{N}^M$ measurements per landmark, the total number of operations is proportional to $\bar{N}^M\bar{M}$, i.e., $\mathcal{O}(\bar{N})$. The Jacobian is calculated during the predictor (EKF) run, which has execution complexity proportional to the window length, K, times the complexity of the measurement update, which is in average proportional to the number of measurements updates. This number will be the average number of measurements per time step times the number of time steps, $\bar{N}^K K$, so the complexity is $\mathcal{O}(\bar{N})$, the same as for the Jacobian. All this together gives that the total computation complexity for calculating the Jacobian in the window will be $\mathcal{O}(\bar{N})$. We see that it scales linearly with the size of the window.

B. SW-NLS-SLAM

SW-NLS-SLAM stacks both the landmarks and platform's motion parameters in the parameter vector, which leads to a Jacobian consisting of the partial derivatives of the residuals with respect to both the landmarks and the motion parameters. In our case, there are both acceleration and angular rates motion parameters of size proportional to K in addition to the landmark measurement residuals. The full analysis of NLS-SLAM is done in [27] for the whole batch, and the corresponding result for a window is $\mathcal{O}(K^2 + \bar{N}K + \bar{N} + K)$. Here, we see that the dominating complexity scales quadratically in window size.

C. GraphSLAM

In the GraphSLAM approach, the Jacobian matrix consists of derivatives for positions and rotations with respect to each other, giving a symmetric matrix with dimension proportional to K^2. The analysis of the complexity for GraphSLAM is done in [27], showing that, on average, the complexity will be $\mathcal{O}(\bar{N}^M K)$.

The complexity comparison between SW-PEM-SLAM and SW-NLS-SLAM is illustrated empirically in Section VI.

V. MOTION AND MEASUREMENT MODELS

In this section, the dynamic and sensor measurements models used for estimation in the visual/inartial SLAM problem are introduced. The used sensors are monocular cameras and 6-DOF inertial sensors, i.e., gyroscopes and accelerometers. The camera and inertial sensors are rigidly coupled to the platform in this setup.

The inertial sensors are here treated as inputs to a dynamic system in order to keep the size of the state as small as possible. Also, a minimal 3D point landmark parametrization is used and its measurement function is given by the pinhole projection model.

Throughout the paper, we will use following frames of reference:

- World frame—global frame for expressing platform's and landmarks' position,
- Navigation frame—same as World frame but translated to platform's position,
- Body frame—frame rigidly attached to the platform's body, same origin as Navigation frame but rotated with the body,
- Camera frame—frame rigidly attached to the camera. In general, this can be different than the Body frame, but in this work, we assume that the Camera and Body frames are the same.

All these frames are Cartesian and locally defined. This implies that platforms' and landmarks' position are only estimated locally in the arbitrary chosen World frame. This is customary approach in SLAM since initial position of the platform is usually unknown.

A. State Dynamics

The gyroscope signals, considered as inputs, are denoted $u^\omega = [u_x^\omega, u_y^\omega, u_z^\omega]^T$, where the subscript refers to each axis of the body frame. Similarly, the accelerometer signals are denoted $u^\mathrm{a} = [u_x^\mathrm{a}, u_y^\mathrm{a}, u_z^\mathrm{a}]^T$. Both of these are measured in the body frame. A discretized dynamic model, where the states are three-dimensional position, velocity, and rotation, $[p_t^T, v_t^T, q_t^T]^T$, in the navigation frame, is then

$$p_{t+1} = p_t + T_s v_t + \frac{T_s^2}{2}\mathrm{R}^T(q_t)(u_t^\mathrm{a} + g^b + w_t^\mathrm{a}), \tag{7a}$$

$$v_{t+1} = v_t + T_s \mathrm{R}^T(q_t)(u_t^\mathrm{a} + g^b + w_t^\mathrm{a}), \tag{7b}$$

$$q_{t+1} = \exp\left(\frac{T_s}{2} S_\omega(u_t^\omega + w_t^\omega)\right) q_t, \tag{7c}$$

where T_s denotes the sampling interval, $\mathrm{R}(q_t)$ is a rotation matrix parametrization of the unit quaternion $q_t = [q_t^0, q_t^1, q_t^2, q_t^3]^T$, which describes the rotation from navigation to body frame is defined as (for each column)

$$\mathrm{R}_{:,1}(q) = \begin{bmatrix} (q^0)^2 + (q^1)^2 - (q^2)^2 - (q^3)^2 \\ 2(q^1 q^2 - q^0 q^3) \\ 2(q^1 q^3 + q^0 q^2) \end{bmatrix}$$

$$\mathrm{R}_{:,2}(q) = \begin{bmatrix} 2(q^1 q^2 + q^0 q^3) \\ (q^0)^2 - (q^1)^2 + (q^2)^2 - (q^3)^2 \\ 2(q^2 q^3 - q^0 q^1) \end{bmatrix}$$

$$\mathrm{R}_{:,3}(q) = \begin{bmatrix} 2(q^1 q^3 - q^0 q^2) \\ 2(q^2 q^3 + q^0 q^1) \\ (q^0)^2 - (q^1)^2 - (q^2)^2 + (q^3)^2 \end{bmatrix}.$$

The gravity expressed in the body frame is $g^b = \mathrm{R}(q_t) g^n$, where $g^n = [0, 0, -g]^T$ is the local gravity vector expressed in the navigation frame, with $g \approx 9.82$, and $\exp(\cdot)$ denotes here the matrix exponential. The noise terms are assumed Gaussian and independent,

$$\left[\left(w_t^\mathrm{a}\right)^T, \left(w_t^\omega\right)^T\right]^T = w_t \sim \mathcal{N}(0, diag(Q_\mathrm{a}, Q_\omega)) = \mathcal{N}(0, Q).$$

For any vector $a = [a_x, a_y, a_z]^T \in \mathbb{R}^3$, the skew-symmetric matrix defined as

$$S_\omega(a) = \begin{bmatrix} 0 & -a_x & -a_y & -a_z \\ a_x & 0 & a_z & -a_y \\ a_y & -a_z & 0 & a_x \\ a_z & a_y & -a_x & 0 \end{bmatrix} \tag{9}$$

is used to parametrize the quaternion dynamics in (7c). It is also worth noticing that, in this case, the dynamics of the system are independent of the landmarks (parameters θ), which simplifies the recursive calculation of the Jacobian.

B. Camera Measurements

The monocular camera is modeled here as a standard pinhole camera, see, cf. [8]. The camera is assumed to be calibrated for its intrinsic parameters (calibration matrix and lens distortion) prior to usage. This enables the usage of the camera as a projective map in Euclidean space, $\mathsf{P} : \mathbb{R}^3 \to \mathbb{R}^2$ by premultiplying the lens undistorted pixel coordinates with the camera calibration matrix. The projection P is defined as

$$\mathsf{P}([X, Y, Z]) = \left[\tfrac{X}{Z}, \tfrac{Y}{Z}\right]^T = [\mathsf{x}_t, \mathsf{y}_t]^T \tag{10}$$

where the Z-coordinate is assumed to be positive and non-zero. A normalized camera measurement, $y_t = [\mathsf{x}_t, \mathsf{y}_t]^T$, of a single landmark, $\theta = [\theta_x, \theta_y, \theta_z]$, at time t is then

$$y_t = \mathsf{P}(\mathsf{R}(q_t)(\theta - p_t)) + e_t, \tag{11}$$

which relates the three-dimensional position and orientation of the camera to the three-dimensional location of the point. The measurement noise is assumed i.i.d. Gaussian, $e_t = [e_t^{\mathsf{x}}, e_t^{\mathsf{y}}]^T \sim \mathcal{N}(0, R)$.

Equation (11) defines a measurement of one landmark at time t. In order to relate all the measurements to a correct landmark, correspondence variables are used. At time t, correspondence variables $\mathsf{C}_t = \{c_t^j\}_{j=1}^{\mathsf{N}_t} \subseteq \{1, \ldots, M\}$, encode the measurement-landmark assignment, $y_t^j \leftrightarrow \theta^{c_t^j}$. As defined before, N_t is the number of measurements and M is the number of all landmarks. This gives that all observed landmarks at time t are defined as a set $\mathsf{M}_t = \{\theta^{c_t^j}\}_{j=1}^{\mathsf{N}_t}$, where $c_t^j = l$ if a measurement j corresponds to a landmark l. The stacked measurement equation, for all observed landmarks at time t, is then

$$\underbrace{\begin{bmatrix} \mathsf{x}_t^1 \\ \mathsf{y}_t^1 \\ \vdots \\ \mathsf{x}_t^{\mathsf{N}_t} \\ \mathsf{y}_t^{\mathsf{N}_t} \end{bmatrix}}_{y_t^{\text{cam}}} = \underbrace{\begin{bmatrix} \mathsf{P}\left(\mathsf{R}(q_t)(\theta^{c_t^1} - p_t)\right) \\ \vdots \\ \mathsf{P}\left(\mathsf{R}(q_t)(\theta^{c_t^{\mathsf{N}_t}} - p_t)\right) \end{bmatrix}}_{h_t(x_t, \mathsf{M}_t)} + \underbrace{\begin{bmatrix} e_t^{1\mathsf{x}} \\ e_t^{1\mathsf{y}} \\ \vdots \\ e_t^{\mathsf{N}_t\mathsf{x}} \\ e_t^{\mathsf{N}_t\mathsf{y}} \end{bmatrix}}_{e_t^{\text{cam}}}, \tag{12}$$

where $e_t^{\text{cam}} \sim \mathcal{N}(0, R_{\text{cam}})$. R_{cam} is a diagonal matrix since all the measurements are assumed to be mutual independent. Note that Θ_i in (4a) consists of union of the landmarks observed in the window i, i.e., $\Theta_i = \bigcup_{t=i}^{i+K} \mathsf{M}_t$. Note that number of elements in Θ_i is M_i.

Solving the correspondence problem in order to find C_t (also known as data association) is outside the scope for this work and is thus assumed solved.

C. Predictor

The nonlinear predictor needed in PEM is realized with an EKF in our case, since it allows for explicit calculation of the gradient of the loss function in (3). The time update from the EKF produces a predicted state estimate at time t given all the measurements up to time $t-1$, $\hat{x}_{t|t-1}$. This prediction together with the measurement model in (11) can be used to obtain the predicted measurement needed for the PEM loss function, namely $\hat{y}_{t|t-1}(\Theta) = h_t(\hat{x}_{t|t-1}(\Theta), \Theta)$ ($\hat{x}_{t|t-1}(\Theta)$ emphasizes the predicted state's dependency on the parameters). The gradient of the PEM loss function (4a) w.r.t. parameters Θ, needed in the solution procedure, is defined as

$$\begin{aligned} \frac{\partial}{\partial \Theta} V(\Theta) &= \frac{1}{2N} \sum_{t=1}^{N} \frac{\partial}{\partial \Theta} \| y_t - h_t(\hat{x}_{t|t-1}(\Theta), \Theta) \|_2^2 \\ &= \frac{1}{N} \sum_{t=1}^{N} J_t(\Theta)^T r_t(\Theta), \end{aligned} \tag{13}$$

where

$$r_t(\Theta) = y_t - h_t(\hat{x}_{t|t-1}(\Theta), \Theta) \tag{14a}$$

$$J_t(\Theta) = \frac{\partial r_t}{\partial \Theta} = -\frac{\partial h_t(x, \Theta)}{\partial x} \frac{\partial x}{\partial \Theta} - \frac{\partial h_t(x, \Theta)}{\partial \Theta}\bigg|_{x = \hat{x}_{t|t-1}(\Theta)}. \tag{14b}$$

The EKF can calculate the explicit value in (13) during the recursive state and measurement updates. This implies that the computational cost for the gradient is (proportionally) linear in the batch length, and consequently scales better than the NLS, which has (proportionally) quadratic cost, [27]. All the details about how the EKF calculations are done are omitted here, and the reader is referred to [26] instead. With the residual $r_t(\Theta)$ and the Jacobian $J_t(\Theta)$ accessible, any NLS solver can be used to estimate the parameter values. In this particular case, a Levenberg–Marquardt solver is used [16], [21].

VI. RESULTS

The performance of the method is evaluated with MC simulations on the synthetic data, as well as with the real data. We have chosen to evaluate the methods on platforms' and landmarks' positions only. This is due to a lack of ground truth for rotations in the case of real data. All the setup and results are described in the subsections below.

A. Synthetic Data

1) Setup: The simulated trajectory consists of 205 camera measurements at 4 Hz and 2050 acceleration and angular rate measurements at 40 Hz, which gives the total trajectory duration time of 1.24 s. This setup is illustrated in Fig. 2. All simulation examples are based on 50 MC simulations, where the noise on the accelerations, angular rates, and camera measurements is varied for each MC run, and it was sampled from the Gaussian distribution with zero mean and standard deviations $\sigma_a = 10^{-3}\text{m/s}^2$, $\sigma_\omega = 10^{-4} \circ \text{s}^{-1}$, and $\sigma_{\text{cam}} = 10^{-4}\,\text{m}$. As previously explained, the correspondence between

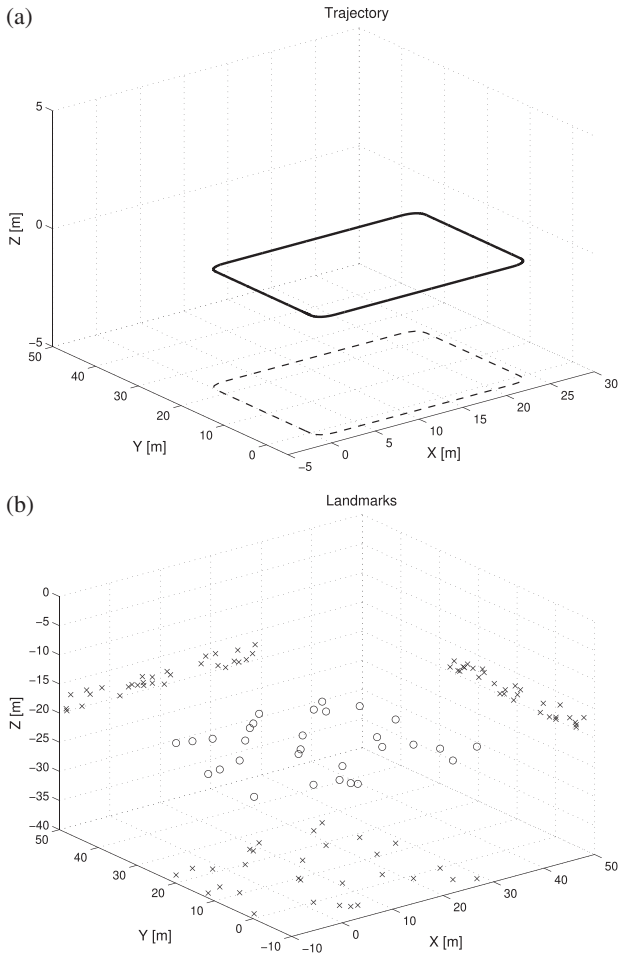


Figure 2. Synthetic environment setup used in MC experiments. (a) Simulated trajectory used in MC simulations with projection on XY-plane for a clearer view (it has constant altitude, i.e., $Z = 0$). (b) Simulated landmarks used in MC simulations with projections on all planes for a clearer view.

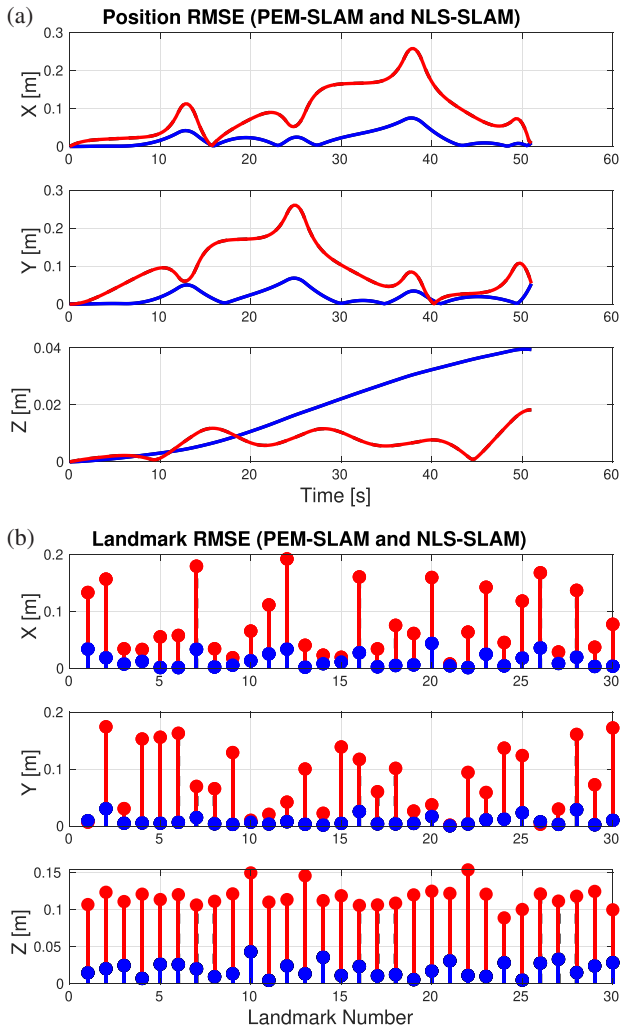


Figure 3. RMS errors for the trajectory and landmarks based on 50 MC simulations for both PEM-SLAM (blue) and NLS-SLAM (red). (a) RMS error for the trajectory estimated with PEM-SLAM (blue) and NLS-SLAM (red) based on 50 MC simulations. (b) RMS error for all the landmarks estimated with PEM-SLAM (blue) and NLS-SLAM (red) for each coordinate based on 50 MC simulations.

measurement and landmarks was known; however, all landmarks were not observed in each camera frame.

2) MC Simulations—Whole batch: The results for the whole batch of data (i.e., "infinite horizon") for both PEM-SLAM and NLS-SLAM are shown in Fig. 3 for the trajectory and the landmarks, respectively.

3) MC Simulations—SW: For the SW approach, 50 MC simulations are done for each of the horizon lengths chosen from {10, 15, 20, 25, 30}. The results, in the form of total position RMSE for the trajectory and some landmarks, for each SW-PEM-SLAM and SW-NLS-SLAM have quite similar and comparable performance, although SW-PEM-SLAM has better RMSE in total for both trajectory and landmarks, just as in the case for the whole batch, see Fig. 4. It is also noticeable that the total error is varying with the horizon length, but not consistently for the SW-PEM-SLAM. It is not necessarily smaller everywhere for the longer horizons, except at the end of the trajectory. SW-NLS-SLAM is more consistent in this regard, at least for the trajectory. Both of the methods are showing this behavior for the landmark position error, i.e., it is not monotonously decreasing with the horizon length for some landmarks, but the general trend is that most landmarks have smaller error for the longer horizon.

B. Real Data

1) Yamaha Rmax: The first real dataset comes from the flight trials performed at Revingehed, Sweden, where a remotely piloted helicopter Yamaha Rmax, see Fig. 5, was flown, [5]. The helicopter was equipped with all the utilized sensors (i.e., IMU and camera). The ground truth (based on GPS) flight trajectory in XY-plane is shown in Fig. 6.

For the validation of the methods, only the horizon length of five images was used, since a short horizon would give a short delay time, which is a realistic assumption for the real-time application. The error between estimated trajectory and GPS- based one for

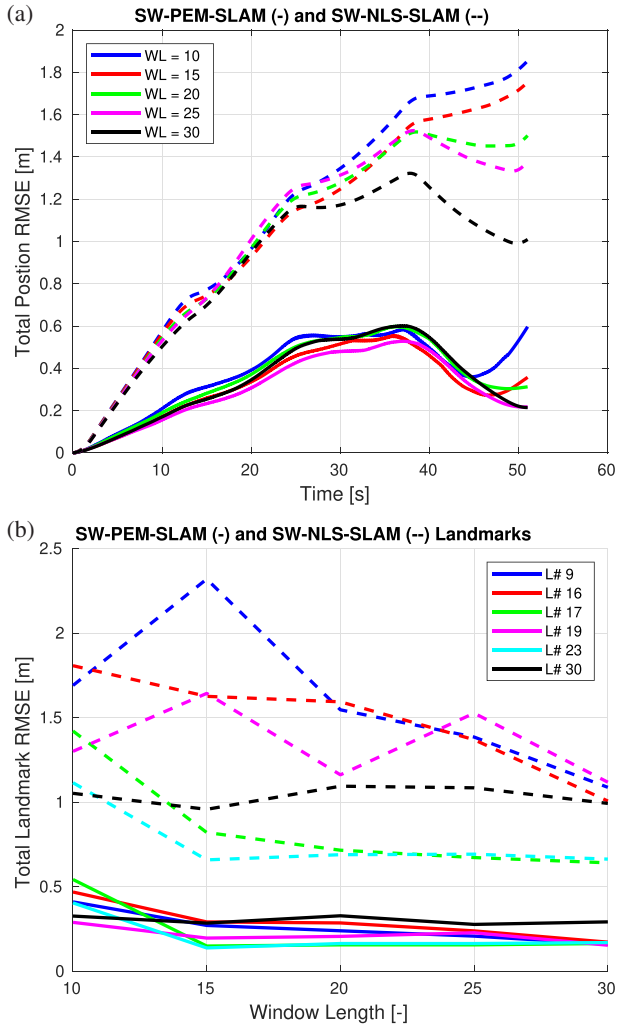


Figure 4. Total RMS errors for the trajectory and seven landmarks for each window length based on 50 MC simulations for SW-PEM-SLAM (solid line) and SW-NLS-SLAM (dashed line). (a) Total RMS error for the trajectory estimated with SW-PEM-SLAM (solid) and SW-NLS-SLAM (dashed) for each window length based on 50 MC simulations. (b) Total RMS error for six landmarks estimated with SW-PEM-SLAM (solid) and SW-NLS-SLAM (dashed) for each window length based on 50 MC simulations.

Figure 5. Remotely piloted helicopter Yamaha Rmax used in the experiments.

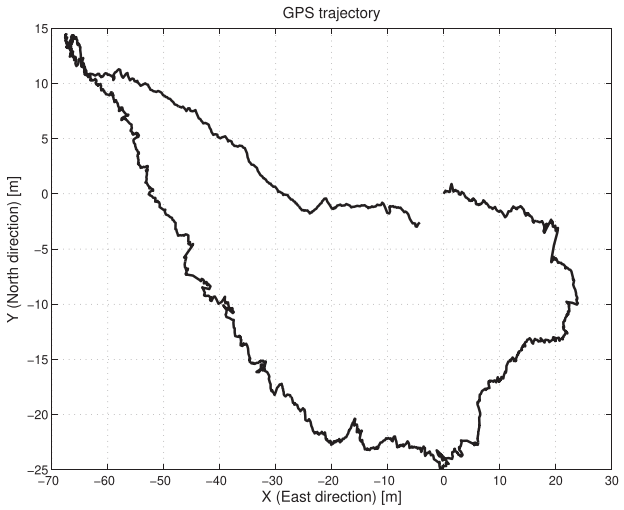


Figure 6. Real data trajectory in XY-plane based on GPS (ground truth).

both SW-PEM-SLAM and SW-NLS-SLAM methods is shown in Fig. 7. Even here, both methods have similar performance for, at least, the X- and Y-coordinates. For the Z-coordinate (or altitude), a much larger error is present. This is a consequence of the inherent visual/inertial SLAM problem property, where it is hard to estimate both distance to landmarks and own localization due to imperfect inertial data. These errors and biases are, in general, not completely removed by the estimation. This can only be done if the inertial data are perfect, which is never the case; see also [22] for further discussion. This behavior is also visible in the simulated data for the Z-coordinate, see, e.g., Fig. 3. In order to remedy this behavior, another kind of stabilizing measurement, like barometric pressure measurements, could be used. Unfortunately, these kinds of measurements were not available. Interestingly, the error is actually decreasing at the end of the trajectory estimated

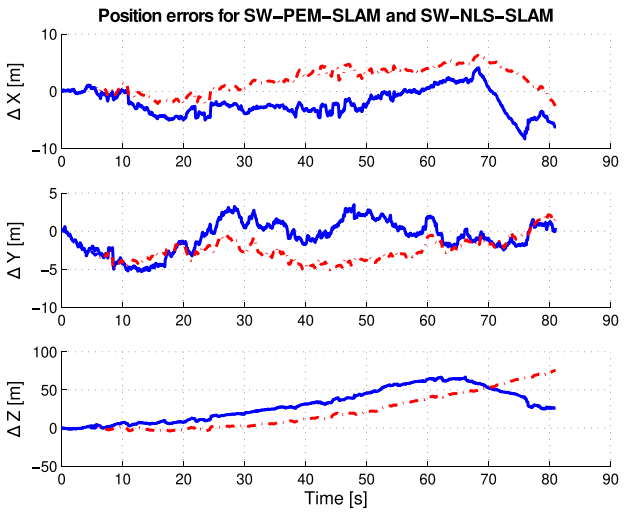


Figure 7. Error between estimated and GPS trajectory for all coordinates for the real data and both SW-PEM-SLAM (blue) and SW-NLS-SLAM (red).

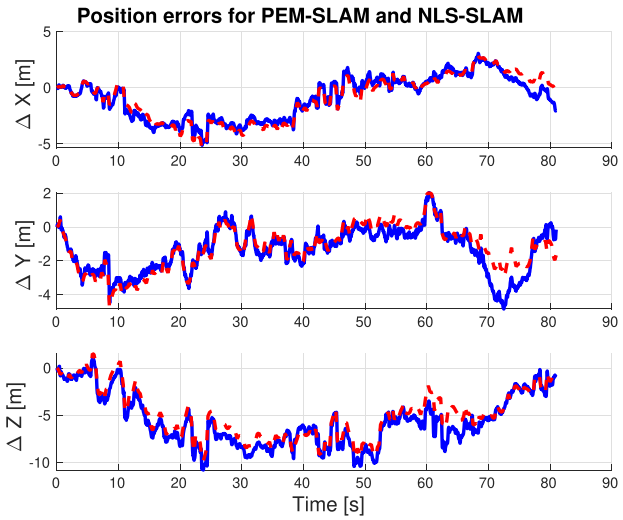


Figure 8. Error between estimated and GPS trajectory for all coordinates for the real data for the whole batch. PEM-SLAM (blue) and NLS-SLAM (red).

by SW-PEM (after 65 seconds approximately). For comparison, the estimate for the whole batch is shown in Fig. 8. Here, the performance is even more similar, and the error in the Z-coordinate is not as prominent (although still larger than X- and Y-coordinates). This is most probably due to the both better loopclosure as well as utilization of the whole data batch instead of only a limited window. Evaluation of map estimation is done by projecting landmarks in an image where they are not observed and comparing these to their measurements from another image where they are observed. This is depicted in Fig. 9. It can be seen that the performance of both methods is quite similar, and that batch estimation has slightly better performance, which is expected.

2) EuRoC: Publicly available datasets from EuRoC MAV [3] have been used in order to evaluate the (SW-) PEM-SLAM performance on another dataset and see how it compares to others' results on the same data. Two out of eleven datasets from EuRoC MAV have been tested. For this dataset, we only show the total RMS error for the whole trajectory and summing all axes as done in [23]. For the SW-PEM-SLAM and dataset MH_01, the RMSE was 0.20 m, and for the PEM-SLAM it was 0.16 m. A horizon of length 20 was used. For the dataset, MH_03 SW-PEM-SLAM had RMSE of 0.32 m with the horizon of length 20. To get the similar performance as for the other dataset, a horizon needed to be 32 long, and the RMSE was 0.21 m in that case. PEM-SLAM had a RMSE of 0.12 m. Compared with the results in, e.g., [23], where RMSE for MH_01 and MH_03 was about 0.07 m, both the SW-PEM-SLAM and PEM-SLAM solutions are in the same order of magnitude. It is worth noting that RMSE in [23] was calculated for keyframes only, while RMSE for SW-PEM-SLAM and PEM-SLAM was for the whole trajectory.

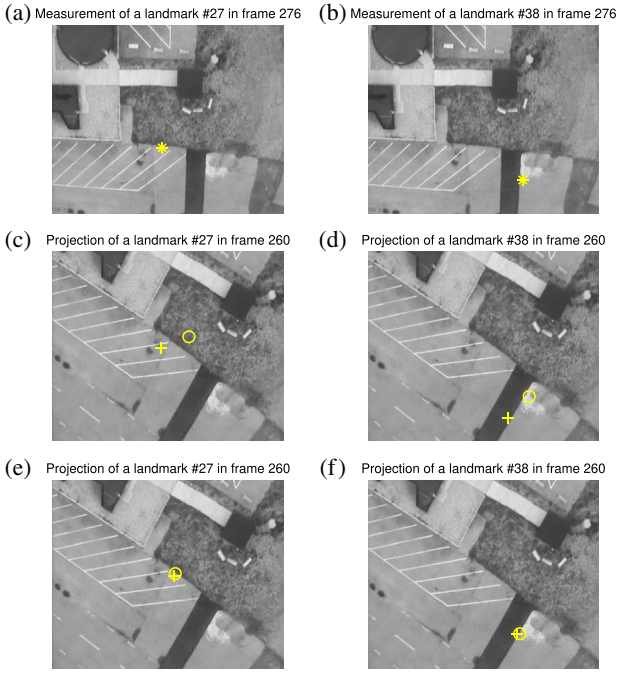


Figure 9. Comparison between measured and reprojected landmarks for the real data and for the SW approach and the whole batch. PEM-SLAM (○) and NLS-SLAM (+) for both SW and batch reprojections. Landmarks are not measured in the images where they are reprojected. (a) Measurement of landmark #27 from image 276. (b) Measurement of landmark #38 from image 276. (c) Reprojection of landmark #27 in image 260 (SW). (d) Reprojection of landmark #38 in image 260 (SW). (e) Reprojection of landmark #27 in image 260 (batch). (f) Reprojection of landmark #38 in image 260 (batch).

C. Execution Time Evaluation

The relative execution time as a function of the horizon length for the SW-PEM-SLAM and the SW-NLS-SLAM are compared. It can be seen in Fig. 10 that the increase for SW-PEM has a linear trend, while for SW-NLS, the trend seems to be quadratic (or at least proportionally quadratic). The plots are produced with the

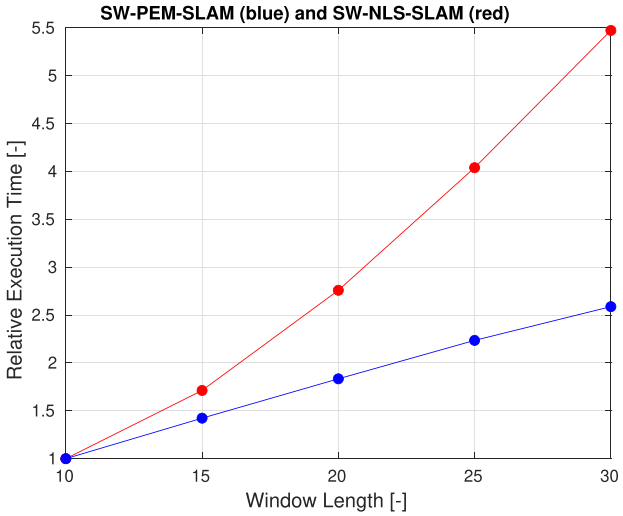


Figure 10. Relative execution time as a function of the horizon length for the SW-PEM-SLAM (blue) and the SW-NLS-SLAM (red).

simulated data averaged over 50 MC runs. Since the estimation accuracy is comparable between the methods, the linear execution time increase for the SW-PEM is a great advantage over SW-NLS when it comes to real-time performance and is one of the main motivations for the choice of SW-PEM-SLAM.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, it is presented how a system identification method, PEM, can be applied to an SLAM problem. This is done by considering the map, here modeled as three-dimensional pointlandmarks, as parameters in the system to be identified, and the motion of the platform observing the landmarks with a monocular camera as dynamic states of the system. Estimation is done in an SW fashion as well as for the whole batch of data. The SW estimation is more appropriate for the real-time adaptation of the estimator, while the whole batch is an offline method. The main advantage of the PEM approach compared to the NLS is the separation between the landmarks and the state estimation, which allows for computation complexity reduction, especially when the horizon length increases. The estimation performance of the SW-PEM-SLAM (and PEM-SLAM, i.e., for the whole batch) is evaluated with MC simulations on both inertial/visual synthetic and real datasets, and compared to SW-NLS-SLAM (and NLS-SLAM) showing comparable performance.

In the continuation of this work, some possible alternative parametrizations of the predictor might be explored, for example, innovation form as in [18]. Also, further development toward better implementation will be pursued.

ACKNOWLEDGMENTS

The authors would like to thank Mariusz Wzorek and Piotr Rudol from AILAB at Linköping University for the Yamaha Rmax data.

REFERENCES

[1] T. Bailey and H. Durrant-Whyte
"Simultaneous localization and mapping (SLAM): Part II,"
IEEE Robot. Automat. Mag., vol. 13, no. 3, pp. 108–117, Sep. 2006.

[2] Stephen Boyd and Lieven Vandenberghe
Convex Optimization. Cambridge, UK: Cambridge University Press, 2004.

[3] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M.W. Achtelik, and R. Siegwart
"The EuRoC micro aerial vehicle datasets,"
Int. J. Robot. Res., vol. 35, no. 10, pp. 1157–1163, 2016.

[4] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós
"ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM,"
IEEE Trans. Robot., vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[5] Patrick Doherty, Patrik Haslum, Fredrik Heintz, Torsten Merz, Per Nyblom, T. Persson, and B. Wingman
"A distributed architecture for autonomous unmanned aerial vehicle experimentation,"
in *Proc. Int. Symp. Distrib. Auton. Robotic Syst.*, 2004, pp. 233–242.

[6] J. Duo, L. Zhao, and J. Mao
"Sliding window based monocular slam using nonlinear optimization,"
in *Proc. 2018 Chin. Intell. Syst. Conf.*, 2019, pp. 519–529.

[7] H. Durrant-Whyte and T. Bailey
"Simultaneous localization and mapping: Part i,"
IEEE Robot.Automat. Mag., vol. 13, no. 12, pp. 99–110, Jun. 2006.

[8] R. I. Hartley and A. Zisserman
Multiple View Geometry in Computer Vision. 2nd ed. Cambridge, U.K.: Cambridge University Press, 2004.

[9] R. Hermann and A. Krener
"Nonlinear controllability and observability,"
IEEE Trans. Autom. Control, vol. 22, no. 5, pp. 728–740, Oct. 1977.

[10] V. Ila, L. Polok, M. Solony, and P. Svoboda
"SLAM—a highly efficient and temporally scalable incremental SLAM framework,"
Int. J. Robot. Res., vol. 36, no. 2, pp. 210–230, 2017.

[11] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert
"iSAM2: Incremental smoothing and mapping using the bayes tree,"
Int. J. Robot. Res., vol. 31, no. 2, pp. 216–235, 2012.

[12] Ayoung Kim and R. Eustice
"Pose-graph visual slam with geometric model selection for autonomous underwater ship hull inspection,"
in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 1559–1565.

[13] M. Kok and T. B. Schön
"Maximum likelihood calibration of a magnetometer using inertial sensors,"
in *Proc. 19th IFAC World Congr.*, 2014, pp. 92–97.

[14] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard
"G^2o: A general framework for graph optimization,"
in *Proc. IEEE Int. Conf. Robot. Automat.* 2011, pp. 3607–3613.

[15] R. Larsson, Z. Sjanic, M. Enqvist, and L. Ljung
"Direct prediction-error identification of unstable nonlinear systems applied to flight test data,"
in *Proc. 15th IFAC Symp. Syst. Identification*, 2009, pp. 144–149.

[16] Kenneth Levenberg
"A method for the solution of certain non-linear problems in least squares,"
Quart. J. Appl. Mathmatics, vol. II, no. 2, pp. 164–168, 1944.

[17] L. Ljung
"Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems,"
IEEE Trans. Autom. Control, vol. 24, no. 1, pp. 36–50, Feb. 1979.

[18] L. Ljung
System Identification, Theory for the User, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1999.

[19] L. Ljung and T. Söderström
Theory and Practice of Recursive Identification, in Signal Processing, Optimization, and Control, vol. 4. Cambridge, MA, USA: MIT Press, 1983.

[20] M. Kaess, A. Ranganathan, and F. Dellaert
"iSAM: Incremental smoothing and mapping,"
IEEE Trans. Robot., vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[21] Donald W. Marquardt
"An algorithm for least-squares estimation of nonlinear parameters,"
SIAM J. Appl. Math., vol. 11, no. 2, pp. 431–441, 1963.

[22] A. Martinelli
"Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination," *IEEE Trans. Robot.* vol. 28, no. 1, pp. 44–60, Feb. 2012.

[23] R. Mur-Artal and J. D. Tardós
"ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[24] R. Mur-Artal and J. D. Tardós
"Visual-inertial monocular SLAM with map reuse," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 796–803, Apr. 2017.

[25] G. Sibley, L. Matthies, and G. Sukhatme
Sliding Window Filter for Incremental SLAM. Boston, MA, USA: Springer, 2008, pp. 103–112.

[26] Z. Sjanic and M. A. Skoglund
"Prediction error method estimation for simultaneous localisation and mapping," in *Proc. 19th Int. Conf. Inf. Fusion*, 2016, pp. 927–934.

[27] Z. Sjanic, M. A. Skoglund, and F. Gustafsson
"EM-SLAM with inertial/visual applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 1, pp. 273–285, Feb. 2017.

[28] Z. Sjanic, M. A. Skoglund, T. B. Schön, and F. Gustafsson
"A nonlinear least-squares approach to the SLAM problem," in *Proc. 18th IFAC World Congr.* 2011, pp. 4759–4764.

[29] Zoran Sjanic and Martin A. Skoglund
"Exploitation of the conditionally linear structure in visual-inertial estimation," in *Proc. 25th Int. Conf. Inf. Fusion*, 2022, pp. 1–8.

[30] T. Söderström and P. Stoica
System Identification. Hemel Hempstead, UK: Prentice-Hall International, 1989.

[31] Hauke Strasdat, J. M. M. Montiel, and Andrew J. Davison
"Scale drift-aware large scale monocular slam," in *Robotics: Science and Systems*, Y. Matsuoka, H. F. Durrant-Whyte, and J. Neira, Eds. Cambridge, MA, USA: The MIT Press, 2010.

[32] H. Wang, S. Huang, K. Khosoussi, U. Frese, G. Dissanayake, and B. Liu
"Dimensionality reduction for point feature SLAM problems with spherical covariance matrices," *Automatica*, vol. 51, pp. 149–157, 2015.

[33] Z. Wang, S. Huang, and G. Dissanayake
"D-SLAM: A decoupled solution to simultaneous localization and mapping," *Int. J. Robot. Res.*, vol. 26, no. 2, pp. 187–204, 2007.

[34] S. Williams, V. Indelman, M. Kaess, R. Roberts, J. Leonard, and F. Dellaert
"Concurrent filtering and smoothing: A parallel architecture for real-time navigation and full smoothing," *Int. J. Robot. Res.*, vol. 33, pp. 1544–1568, 2014.

[35] L. Zhao, S. Huang, and G. Dissanayake
"Linear SLAM: Linearising the SLAM problems using submap joining," *Automatica*, vol. 100, pp. 231–246, 2019.

Zoran Sjanic (MSc-01, PhD-13) has been employed by Saab Aeronautics, Linköping, Sweden, since 2001, where he is working with the Department of Sensor Fusion and Tactical Control as a System Engineer and Technical Manager for the image analysis and processing system in Gripen fighter aircraft. He has also worked as a Technical Manager for the Navigation System in both Gripen and Skeldar UAS. He also holds a position as an Adjunct Associate Professor in the Division of Automatic Control, Department of Electrical Engineering, at Linköping University, Linköping, Sweden.

His research interests include sensor fusion for navigation of manned and unmanned aircraft, target tracking, simultaneous localization and mapping, and estimation methods for linear and nonlinear systems.

Martin Skoglund (MSc-08, PhD-14) is an Adjunct Associate Professor in the Division of Automatic Control, Department of Electrical Engineering, Linköping University, Linköping, Sweden. Since 2016, he has been working at the Eriksholm Research Centre, Snekkersten, Denmark, part of Oticon, as a Senior Scientist. His research interests include sensor fusion, modeling, and estimation of nonlinear systems. He works with many types of sensors, such as, vision, inertial, audio, EEG, electromagnetic, and more, with the purpose of finding solutions for future hearing care technology.

Data Association With Camera Parameters Estimation for Object Tracking From Drones

ZIJIAO TIAN
YAAKOV BAR-SHALOM
RONG YANG
HONG'AN JACK HUANG
GEE WAH NG

This paper considers the problem of inaccurate measurement-to-track association (M2TA) and poor tracking caused by camera motion changes in drone-captured video. The camera often changes its field of view to track targets; however, the sudden change leads to inaccurate M2TA and degrades tracking performance. Previous work estimated the 3D camera motion parameter vector (zoom ratio, panning, and tilting) and associated measurements and tracks only between two consecutive frames. This paper extends the camera motion parameter to 4D by including rolling and sequentially associates (forward) measurements to tracks over the entire data. The estimated camera parameters improve the predicted measurements and achieve better M2TA. Results on real data illustrate the benefits of the proposed method (association with 4D camera parameters estimation) that yields better associations and improves tracking accuracy compared to the state-of-the-art gating method based on inflated covariances.

Manuscript received December 26, 2023; revised August 21, 2024; released for publication October 14, 2024

Z. Tian and Y. Bar-Shalom are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269, USA (e-mail: zijiao.tian@uconn.edu; yaakov.bar-shalom@uconn.edu).

R. Yang and H. A. J. Huang are with the DSO National Laboratories, Singapore 118225 (e-mail: yrong@dso.org.sg; hhongan@dso.org.sg).

G. W. Ng is a staff from DSO National Laboratories Singapore 118225, who is on secondment to Home Team Science and Technology Agency, Singapore 138507 (e-mail: ng_gee_wah@htx.gov.sg).

1557-6418/2024/$17.00

I. INTRODUCTION

Unmanned aerial systems (UAS), equipped with cameras, are extensively used to capture images and videos for tracking systems. These systems are crucial for surveillance applications. For example, UAS can be employed to monitor borders to detect illegal crossings or smuggling activities, as well as to observe traffic flow to identify and respond to accidents. These cameras often adjust their field of view (FoV) to keep up with moving targets [12], [16]–[18], [24]. However, changes in pointing and/or in image scale make the target data association and tracking from drones more challenging than traditional data association and object tracking. The camera movements, such as pan (yaw), tilt (pitch), zoom, and platform roll, are not available to the data association and tracking algorithms and can degrade the reliability of the video tracks [7], [15], [19].

Camera vibrations or movements will introduce instability into video images, posing challenges in video image stabilization and registration. Most methods address this problem by compensating camera motion with camera motion estimation. Typically, camera motion estimation is classified into two categories: intensity-based motion estimation and feature-based motion estimation. The intensity-based motion estimation (such as using image grayscale [14], phase correlation [10], pixel-based correlation [11]) is based on pixel intensities between consecutive frames, while the feature-based motion estimation (such as edges and corners [6]) is based on extracting and matching features across consecutive frames. Some advanced video stabilization techniques combine both approaches in [28].

In [23], it was shown that the tilt, pan, and roll angle errors of the camera can affect the navigational parameters in autonomous vehicles. Allebosch et al. [1] compensated camera motion by estimating panning and tilting with different models. A reversible jump Markov chain Monte Carlo method was employed for estimating camera parameters consisting of angles and positions in [8]. Our previous work [25] assumes that the camera motion parameters are described by zoom ratio, panning, and tilting of the focal-plane array. The 3D camera parameter vector is directly solved by the MLE3 (maximum likelihood estimation in 3D) approach via linear least squares (LLS). However, this does not account for UAS camera rolling. Moreover, it was tested only between two consecutive frames and was not applied to the entire duration of the data [25]. Thus, it could not provide an evaluation of the tracking accuracy during an entire video sequence.

In this work, the goal is to develop an approach for accurate tracking in the presence of camera panning, tilting, zooming, and rolling for drone-captured video. Targets are detected by a state-of-the-art object detection algorithm—You Only Look Once (YOLO) [20], which provides bounding boxes (BBs). To track accurately, the camera has the capability to adjust its view,

and the UAS can change its velocity and altitude. This is accomplished visually by a human operator. The attitude is operator-controlled but not quantified to be usable for the algorithm. The camera motion parameters, unknown to the operator or the tracker, are represented by a 4D vector including zoom ratio, panning, tilting, and rolling. These parameters are estimated using the MLE4 (maximum likelihood estimation in 4D) via the iterated least squares (ILS) method in each frame. Based on the estimated camera parameters, one associates the measurement-track pairs, and then corrected state predictions are calculated. These corrected state predictions play an important role in the filtering step (Kalman filter), resulting in more accurate state updates. Compared with the conventional measurement-to-track association (M2TA), the validation gating method, which relies on inflated measurement covariance for association, the proposed algorithm has superior robustness and performance. If there are unexpected target state changes due to camera movements, the measurements may not be correctly associated with their tracks by the gating method. In contrast, our method has enhanced robustness by integrating camera motion parameter estimation into the association and tracking process. This integration allows our algorithm to maintain accurate tracking even in the presence of abrupt camera motion changes, which would typically challenge the conventional gating method.

The contributions of this paper are as follows: Firstly, it extends the camera motion parameter vector from 3D to 4D, enhancing its capability not only for zoom, pan, and tilt, but also for roll movements. The 3D parameters are estimated using MLE3, while the 4D parameters are estimated using MLE4. The latter is shown to have smaller errors in tracking results compared to 3D parameter vector estimation. Additionally, the proposed method can handle scenarios with both a large and small number of targets with different assignment methods. It adapts effectively to diverse tracking environments.

The rest of the paper is structured as follows: Section II introduces the target detection by YOLO, presents the baseline Gating Method with Inflated Covariance (GMIC) method, and outlines the overall system flow. Section III presents the estimation of the camera motion parameters consisting of both the 3D vector and the 4D vector. Section IV presents the proposed approach that integrates camera parameters into the association process. Section V shows the real data results and discusses them. Section VI draws the conclusions.

II. PROBLEM FORMULATION

This section first introduces the target position detection from YOLO in each frame. Then the baseline M2TA method is briefly presented, which is used to compare with the proposed method. The tracking system flow chart is later shown in this section. After target detection by YOLO, the association is combined with camera parameter estimates to yield better target state predictions.

A. YOLO: Target Detection

A review of deep learning applied to computer vision for target detection can be found in [29]. One of the most popular algorithms is YOLO [20]. YOLO integrates feature extraction, object localization, BB regression, and classification in a monolithic network. It maps from image pixels to BB coordinates and class probabilities. The basic idea is dividing the input image into a grid, and each grid cell is responsible for predicting the object. It uses a single-stage architecture to make predictions for multiple objects, making it faster and more efficient than traditional object detection algorithms. YOLO v3 [21], applied on a per frame basis, is an incremental improvement over YOLO in detection and BB accuracy for smaller targets. Besides, its efficiency in processing time makes it widely used in real-time object detection.

In this paper, the targets are detected by YOLO v3 in each frame and the measurements are the target's positions, specifically, the top left corner of the BB. Although it can identify multiple objects from a video frame and label them with corresponding class probabilities, we focus on people and their 2D position information.

B. Gating Method With Inflated Covariance (GMIC)

The traditional state-of-the-art M2TA method—validation gate method—assumes that the target motion can be utilized to predict the "measurement association regions" [3], [4], [27]. It eliminates unlikely measurements that need to be considered for association with a track [22], [26]. The gating method is based on inflated covariances. The camera parameter changes create bias in the estimation. Instead of estimating the bias, this approach increases the measurement noise standard deviations to "cover" the bias implicitly. For the association, the association gate is enlarged by an inflated measurement error variance so that shifted measurements can be associated with their tracks.

Consider a set of measurements $z_i(k)$ at time (frame index) k, $i = 1, 2, \ldots, N_m(k)$, where $N_m(k)$ is the number of measurements at time k, and a set of predicted measurements $\hat{z}_j(k|k-1)$ at time k, $j = 1, 2, \ldots, N_t(k)$, where $N_t(k)$ is the number of tracks for which a prediction at time k is available. The difference between each actual and predicted measurement (the filter innovation) is defined as

$$\tilde{n}_{ij}(k) = z_i(k) - \hat{z}_j(k|k-1). \tag{1}$$

A gate is formed about the predicted measurement, and all actual measurements (observations) that fall within the gate are considered for track updates. Define a gate threshold γ such that association is allowed if the norm

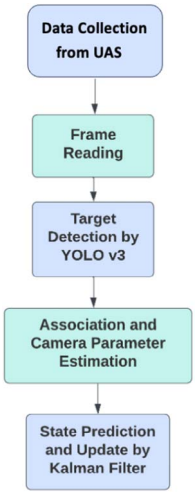


Figure 1. Flow chart of the tracking system.

of the residual falls within the gate

$$d_{ij}^2(k) = \tilde{n}'_{ij}(k) S_j(k)^{-1} \tilde{n}_{ij}(k) \leq \gamma, \tag{2}$$

where d_{ji}^2 is also known as the squared Mahalanobis distance between track j and measurement i and $S_j(k)$ is the corresponding innovation covariance. Once the potential measurements are chosen based on (2), they are associated to tracks using the Auction method. In this process, each track "bids" for the measurements, and the goal is to find the highest "bid" track and then assign the measurement to the track. The details can be found in [4].

In target tracking, the assigned measurements are incorporated into the updated track state estimates during the filtering step. However, it is important to note that gating, while a commonly used heuristic method, is not infallible. Due to sudden camera movements such as panning, zooming, tilting, or rolling, the validation gates may lead to incorrect M2TA and poor tracking performance, even in the presence of inflated gates.

C. System Flow for Assignment With Camera Parameter Estimation

The overall system is illustrated in the flow chart in Fig. 1. It outlines several key steps, including data collection, target detection, a novel approach to association that incorporates camera parameter estimation, and filtering. Compared with the previous gating method, the main difference is that we integrate camera parameter estimation into the association process. This mitigates the effects of sudden camera movements, thus yielding more accurate M2TA and better state prediction. The proposed algorithm sequentially estimates the camera state and the target states at each time step, which is similar to the simultaneous localization and mapping (SLAM) [5]. However, SLAM typically focuses on building a map in a static environment, while our problem focuses on tracking targets whose locations in the FoV change due to camera parameter variation. Besides, our approach has to be implementable in real time with modest computing requirements; adding moving targets as in SLAM would significantly increase these requirements.

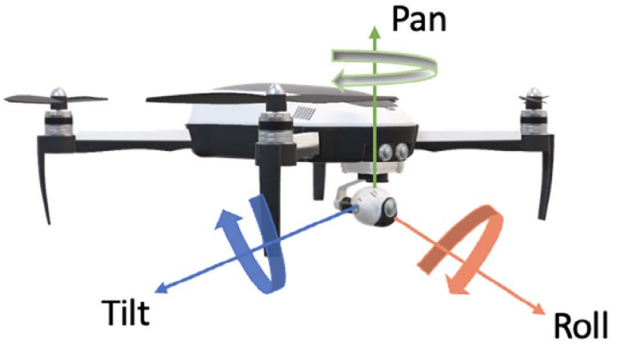


Figure 2. Camera has pan, tilt, and roll movements.

As shown in Fig. 2, the camera mounted on drone gimbals can be controlled by servo driver modules. When it comes to recording of image frames, the less vibration and camera shake the better. However, sudden camera movements are inevitable in practical scenarios. Specifically, we investigate the several camera motion parameters:

1. Pan (Yaw): A yaw motion describes the left and right (horizontal) movement of the camera.

2. Tilt (Pitch): An up and down (vertical) movement of the camera.

3. Roll: A roll motion is a rotation around the camera axis direction. The roll comes from "banking" when the UAS turns.

It should be pointed out that the commands for the above are given usually by a human operator, but they are not measured or available to the data association or tracking algorithms.

Additionally, the camera's field of view can be adjusted by zooming in or out. Although zoom is not a camera position movement, one can change its focal length to change image size. These camera motion parameters are crucial for the quality of tracking and will be estimated in the subsequent section.

III. CAMERA MOTION PARAMETERS ESTIMATION

This section presents the formulation and estimation of the camera parameters using 3D and 4D vectors when the camera has sudden motion change. The corrected target position prediction conditioned on the parameter vector estimates is also given.

A. 3D Camera Parameters Estimation (MLE3)

The 3D camera parameter vector consisting of the zoom ratio $\phi(k)$, camera panning (horizontal motion) $x_c(k)$, and tilting (vertical motion) $y_c(k)$ is given by

$$\zeta(k) = \begin{bmatrix} \phi(k) \\ x_c(k) \\ y_c(k) \end{bmatrix}. \tag{3}$$

The camera parameter vector (3) [or (5) below] is assumed an unknown constant for each k, which is the model for the least squares [2] (LS) algorithm. No dynamics are assumed across time since the goal was simplicity. Using a dynamic model for the camera motion (and a Kalman filter to estimate it) is possible, but it would increase the computation complexity, and since the LS worked very well, there was no need to increase the complexity.

Assume there is a set of measurements $\mathbf{z}_i(k)$ at time $k, i = 1, 2, \ldots, N_m$, and a set of predicted measurements $\hat{\mathbf{z}}_j(k|k-1)$ at time k, $j = 1, 2, \ldots, N_t$. The corrected prediction (denoted by hat and superscript "κ") conditioned on the 3D parameter vector is[1]

$$\hat{\mathbf{z}}_j^{\kappa}[k|k-1, \boldsymbol{\zeta}(k)] = \begin{bmatrix} \hat{x}_j(k|k-1)\phi(k) + x_c(k) \\ \hat{y}_j(k|k-1)\phi(k) + y_c(k) \end{bmatrix}, \quad (4)$$

where $\hat{x}$ and $\hat{y}$ are the predicted position of the target. The 3D camera parameter vector without rolling is solved by MLE (designated as MLE3) via LLS. The details are shown, for completeness, in Appendix A (based on [25]).

B. 4D Camera Parameter Estimation (MLE4)

The 4D camera parameter vector including the rolling $\rho(k)$ is denoted as

$$\boldsymbol{\xi}(k) = \begin{bmatrix} \rho(k) \\ \phi(k) \\ x_c(k) \\ y_c(k) \end{bmatrix}, \quad (5)$$

where ϕ is the zoom ratio, which can be expressed as the ratio of focal length $f(k)$ at different time as follows[2]:

$$\phi(k) = \frac{f(k)}{f(k-1)}. \quad (6)$$

Then, the corrected prediction conditioned on the 4D camera parameter vector is given by

$$\hat{\mathbf{z}}_j^{\kappa}[k|k-1, \boldsymbol{\xi}(k)] = \begin{bmatrix} [\hat{x}_j(k|k-1)\cos\rho(k) + \hat{y}_j(k|k-1)\sin\rho(k)]\phi(k) + x_c(k) \\ [\hat{y}_j(k|k-1)\cos\rho(k) - \hat{x}_j(k|k-1)\sin\rho(k)]\phi(k) + y_c(k) \end{bmatrix}. \quad (7)$$

Define $i(j)$ as the index of the measurement associated with track j. The measured target position is denoted by

$$\mathbf{z}_{i(j)}(k) = \begin{bmatrix} h_1[\hat{x}_{i(j)}(k|k-1), \hat{y}_{i(j)}(k|k-1), \boldsymbol{\xi}(k)] + n_{i(j),1}(k) \\ h_2[\hat{y}_{i(j)}(k|k-1), \hat{x}_{i(j)}(k|k-1), \boldsymbol{\xi}(k)] + n_{i(j),2}(k) \end{bmatrix}, \quad (8)$$

with

$$h_1 = [\hat{x}_{i(j)}(k|k-1)\cos\rho(k) + \hat{y}_{i(j)}(k|k-1)\sin\rho(k)]\phi(k) + x_c(k), \quad (9)$$

$$h_2 = [\hat{y}_{i(j)}(k|k-1)\cos\rho(k) - \hat{x}_{i(j)}(k|k-1)\sin\rho(k)]\phi(k) + y_c(k), \quad (10)$$

where $n_{i(j),\ell}(k), \ell = 1, 2$ are mutually independent zero-mean white Gaussian residuals with variance σ^2. The 4D camera parameters are estimated by the MLE4 algorithm via ILS estimator [2]. The ILS recursion is as follows: Using a first order series expansion about $\hat{\boldsymbol{\xi}}$, one has

$$\mathbf{z}_{i(j)}(k) = \begin{bmatrix} h_1[\hat{x}_{i(j)}(k|k-1), \hat{y}_{i(j)}(k|k-1), \hat{\boldsymbol{\xi}}(k)] \\ h_2[\hat{y}_{i(j)}(k|k-1), \hat{x}_{i(j)}(k|k-1), \hat{\boldsymbol{\xi}}(k))] \end{bmatrix} \quad (11)$$

$$+ \begin{bmatrix} \boldsymbol{J}_1(\hat{\boldsymbol{\xi}}(k) - \hat{\boldsymbol{\xi}}(k+1)) + n_{i(j),1}(k) \\ \boldsymbol{J}_2(\hat{\boldsymbol{\xi}}(k) - \hat{\boldsymbol{\xi}}(k+1)) + n_{i(j),2}(k) \end{bmatrix}. \quad (12)$$

We can define the following matrices:

$$\mathbf{h}_{i(j)} = [h_1 \;\; h_2]' \quad (2 \times 1), \quad (13)$$

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_N \end{bmatrix} \quad (2N \times 1), \quad (14)$$

$$\Re_{i(j)} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \quad (2 \times 2), \quad (15)$$

$$\mathbf{R} = \text{diag}[\Re_1 \ldots \Re_N] \quad (2N \times 2N), \quad (16)$$

$$\mathbf{z}_{i(j)} = [x_{i(j)} \;\; y_{i(j)}]' \quad (2 \times 1), \quad (17)$$

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_N \end{bmatrix} \quad (2N \times 1), \quad (18)$$

where N is the number of M2TA pairings, and $i(j) = 1, \ldots, N$. The Jacobian matrix $\mathbf{J}$ is given by

$$\mathbf{J} = [\mathbf{H}_1 \;\; \mathbf{H}_2 \;\; \ldots \;\; \mathbf{H}_N]' \quad (2N \times 4), \quad (19)$$

where

$$\mathbf{H}_j = \begin{bmatrix} \frac{\partial h_1}{\partial \rho} & \frac{\partial h_1}{\partial \phi} & \frac{\partial h_1}{\partial x_c} & \frac{\partial h_1}{\partial y_c} \\ \frac{\partial h_2}{\partial \rho} & \frac{\partial h_2}{\partial \phi} & \frac{\partial h_2}{\partial x_c} & \frac{\partial h_2}{\partial y_c} \end{bmatrix} \quad (2 \times 4). \quad (20)$$

The partial derivatives are shown in Appendix B.

Finally, the updated ILS estimates $\hat{\boldsymbol{\xi}}(k+1)$ (4×1) is then obtained as

$$\hat{\boldsymbol{\xi}}(k+1) = \hat{\boldsymbol{\xi}}(k) + (\boldsymbol{J}'(k)\boldsymbol{R}^{-1}\boldsymbol{J}(k))^{-1}\boldsymbol{J}'(k)\boldsymbol{R}^{-1}[\boldsymbol{z}(k) - \boldsymbol{h}(\hat{\boldsymbol{\xi}}(k))]. \quad (21)$$

[1]Note that (4) is written with the yet to be estimated camera parameters.

[2]The focal lengths are unknown and not observable (unless the sizes of the targets are known and the noises are much smaller). However, the ratio (6) can be estimated together with the association.

IV. ASSOCIATION WITH CAMERA PARAMETER ESTIMATION

This section first introduces the target motion model. Next, the association with estimation methods in terms of the number of M2TA pairs is presented. The filtering step using the Kalman filter is also provided.

A. Dynamic Models

There are multiple targets in the observed frame, and they are assumed to move with a nearly constant velocity (NCV). The target motion model is characterized by a continuous white-noise acceleration (CWNA) model [2]. The state vector consisting of position and velocity in the camera image is

$$\mathbf{x}(k) = [x(k)\ \ y(k)\ \ \dot{x}(k)\ \ \dot{y}(k)]'. \tag{22}$$

For sampling interval T, the state and measurement equations are

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{v}(k), \tag{23}$$

$$\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{w}(k), \tag{24}$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{25}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{26}$$

The measurements consist of the state's position components. For the above, $\mathbf{v}(k)$ is the zero mean white process noise sequence with covariance

$$\mathbf{Q}(k) = \begin{bmatrix} \frac{T^3}{3} & 0 & \frac{T^2}{2} & 0 \\ 0 & \frac{T^3}{3} & 0 & \frac{T^2}{2} \\ \frac{T^2}{2} & 0 & T & 0 \\ 0 & \frac{T^2}{2} & 0 & T \end{bmatrix} q, \tag{27}$$

where q is the process noise power spectral density (assumed the same in x and y) and $\mathbf{w}(k)$ is the zero mean white measurement noise sequence with covariance $\mathbf{R}(k) = \text{diag}[\sigma^2\ \sigma^2]$.

B. Association With Estimation

The association should be done between the following:

1. Tracks represented by corrected predictions $\hat{\mathbf{z}}_j^{\kappa}[k|k-1, \boldsymbol{\zeta}(k)]$, $j = 0, 1, \ldots, N_t$ (with $\boldsymbol{\zeta}$ or $\boldsymbol{\xi}$ to be estimated),[3] where the index $j = 0$ represents the "dummy tracks" to which the unassociated measurements belong.

2. Measurements $\mathbf{z}_i(k)$, $i = 0, 1, \ldots, N_m$, where the index $i = 0$ represents the "dummy measurements" to which the unassociated tracks belong.

The cost of assigning $\mathbf{z}_i(k)$ to $\hat{\mathbf{z}}_j^{\kappa}[k|k-1, \boldsymbol{\zeta}(k)]$ is the negative log-likelihood function [2] (scalar normalized squared distance)[4]

$$c[i, j, k, \boldsymbol{\zeta}(k)] = ||\mathbf{z}_i(k) - \hat{\mathbf{z}}_j^{\kappa}(k|k-1)||^2. \tag{28}$$

Initial candidate measurement-to-track pairs are based on the GMIC method. There are two different association methods in terms of the number of measurement-to-track pairs:

Method A: When the number of pairs is small, the first iteration estimates camera parameters is based on the first set of measurement-to-track pairs. Then it corrects the predictions. The second iteration of the assignment is based on other pairs of measurements and tracks. The iteration is stopped until all combinations are exhausted. Finally, we choose the assignment that yields the lowest cost along with its estimated parameters.

This method can find the optimal measurement-to-track pairs, but requires a global minimization (exhaustive) search of all the combinations of measurements and tracks. It requires $O(n!)$ operations, where n is the number of the measurement-to-track pairs. Due to this complexity, it is practical only for a limited number of pairs. Typically this is feasible for $n < 6$.

Method B: To enhance computational efficiency for $n \geq 6$, the estimation is combined with the 2D assignment algorithm, specifically the Auction or Hungarian method[5] [9]. The procedure is as follows: The 2D assignment algorithm first finds the assignment of measurements to tracks that minimizes the total cost. Then the camera parameter vector is estimated based on this assignment, the predictions are corrected, and the target state is updated. This method requires $O(n^3)$ running time, making it efficient for a large number of tracks and measurements.

To combine the above methods, when the number of pairs is small ($n! \leq n^3$, for $n < 6$), the first estimation method with global search is utilized. Otherwise, when dealing with a large number of pairs, the estimation with the 2D assignment algorithm is preferred to ensure computational efficiency.

[3]The camera parameter estimation for $\boldsymbol{\zeta}$(3D) or $\boldsymbol{\xi}$(4D) is the same except for their dimensions, so we use the notation $\boldsymbol{\zeta}$ in Section IV.

[4]We assume the innovation covariances are all diagonal and equal, thus we can omit them.

[5]This article focuses on the camera estimation rather than the assignment algorithms. The chosen assignment algorithms are simple for real-time implementation, but one can also use other 2D assignment algorithms.

Table I
The Algorithms Considered in the Paper

Acronym	Algorithm
GMIC	Gating method with inflated covariances
MLE3 [25]	ML estimator for 3D camera parameter vector via LLS
MLE4	ML estimator for 4D camera parameter vector via ILS

C. Predictions

After association and camera parameter estimation, the assigned measurements are corrected and used to obtain the updated track state estimates during the filtering stage. The improved (position) prediction $\hat{\mathbf{z}}_j^{\kappa}[k|k-1, \hat{\boldsymbol{\zeta}}(k)]$ is used for the target state update in the Kalman filter. Thus, the updated state estimate $\hat{\mathbf{x}}$ for target j and the updated covariance are given by[6]

$$\hat{\mathbf{x}}_j(k|k) = \hat{\mathbf{x}}_j^{\kappa}[k|k-1, \hat{\boldsymbol{\zeta}}(k)] + \mathbf{K}(k)\boldsymbol{\nu}_j(k), \tag{29}$$

$$\mathbf{P}_j(k|k) = \mathbf{P}_j(k|k-1) - \mathbf{K}(k)\mathbf{S}(k)\mathbf{K}'(k), \tag{30}$$

where the filter gain $\mathbf{K}(k)$ and innovation covariance are

$$\mathbf{K}(k) = \mathbf{P}_j(k|k-1)\mathbf{H}'(k)\mathbf{S}^{-1}(k), \tag{31}$$

$$\mathbf{S}(k) = \mathbf{H}(k)\mathbf{P}_j(k|k-1)\mathbf{H}'(k) + \mathbf{R}(k), \tag{32}$$

the measurement residual (innovation) is

$$\boldsymbol{\nu}_j(k) = \mathbf{z}_j(k) - \hat{\mathbf{z}}_j^{\kappa}[k|k-1, \hat{\boldsymbol{\zeta}}(k)]. \tag{33}$$

Since the proposed method (2D assignment with camera parameter estimation) will be shown to provide a more accurate prediction, the tracking results will be better than the validation GMIC method when the camera is panning, tilting, zooming or rolling.

V. REAL DATA RESULTS

Two types of real scenarios are considered: with a small number of targets (using exhaustive search) and with a large number of targets (using Hungarian assignment algorithm). MLE4 is compared with MLE3 and GMIC, see Table I.

A. Initialization

The initial candidate associations for both MLE3 and MLE4 are based on GMIC. GMIC merges the nearby YOLO measurements when there is a one-pixel difference. If the merging distance is large, GMIC will make wrong associations, especially for tracks lacking measurements. To get a better association, we only focus on the confirmed tracks, and remove the inactive tracks (for example, previous tracks that moved out of FoV).

[6]The prediction $\hat{\mathbf{x}}_j^{\kappa}[k|k-1, \hat{\boldsymbol{\zeta}}(k)]$ has position components $\hat{\mathbf{z}}_j^{\kappa}[k|k-1, \hat{\boldsymbol{\zeta}}(k)]$, and its velocity components are unchanged from $\hat{\mathbf{x}}_j(k|k-1)$.

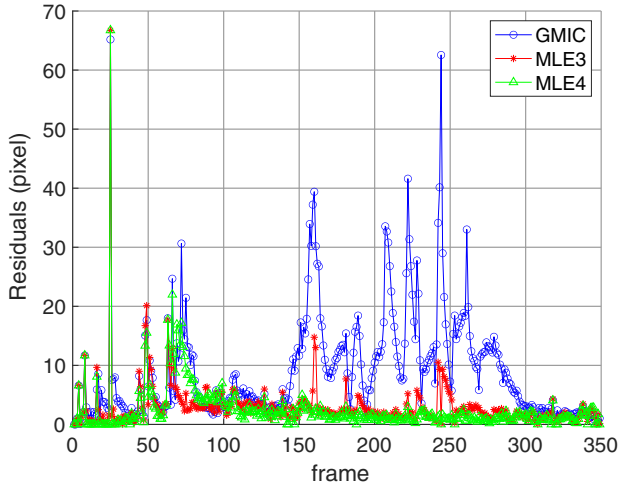


Figure 3. HTX scenario: Average track residual errors for GMIC, MLE3, and MLE4 are 9.04, 2.94, and 2.48, respectively.

The new tracks are initialized such that the first estimate $\hat{\mathbf{x}}_j(k|k)$ equals the observed measurements and velocity zero (one point initialization), and the initial position standard deviation (s.d.) is 0.3b (px) in each component and the initial velocity s.d. is 3b (px/s) where b is the BB width.

For the ILS search of MLE4, the initial 4D camera parameter vector is set as $[0\ 1\ 0\ 0]'$ in equation (5). Based on the 2D assignment algorithm, the optimal 4D parameter vector is estimated by the ILS algorithm (see, e,g., [2]).

B. Small Number of Targets—HTX Video, Assignment With Exhaustive Search

In this section, we examine the scenario considered in [25] with a small number of targets.[7] The UAS only changes its position at a fixed altitude during recording. The sampling frequency is 30 Hz (30 frames/s). The frame has a size of 1920×1080 px. The width of the BBs is around 15 px and the height of the BBs is around 42 px. The average velocity of the BBs is around 23 px/s. Thus, we choose the process noise power spectral density as[8] $q = 16$ (px^2/s^3) and the measurement noise covariance matrix as $R = \text{diag}[9\ 9]$ (px^2).

The performance of tracking accuracy is evaluated by the *average track residual errors*—average track residual errors (ATRE) at each frame, as follows:

$$e(k) = \frac{1}{N_t(k)} \sum_{j=1}^{N_t(k)} \sqrt{\tilde{x}(k)_{i(j)}^2 + \tilde{y}(k)_{i(j)}^2}. \tag{34}$$

where $\tilde{x}$ and $\tilde{y}$ are innovation (residual) errors (the difference between the corrected predicted positions and

[7]https://github.com/zijiaoTian58/HTX_Dataset.
[8]The root mean square (RMS) change in the velocity over a sampling interval T is $\sqrt{qT}$, which for $T = 1/30$ s, yields 0.7 px/s.

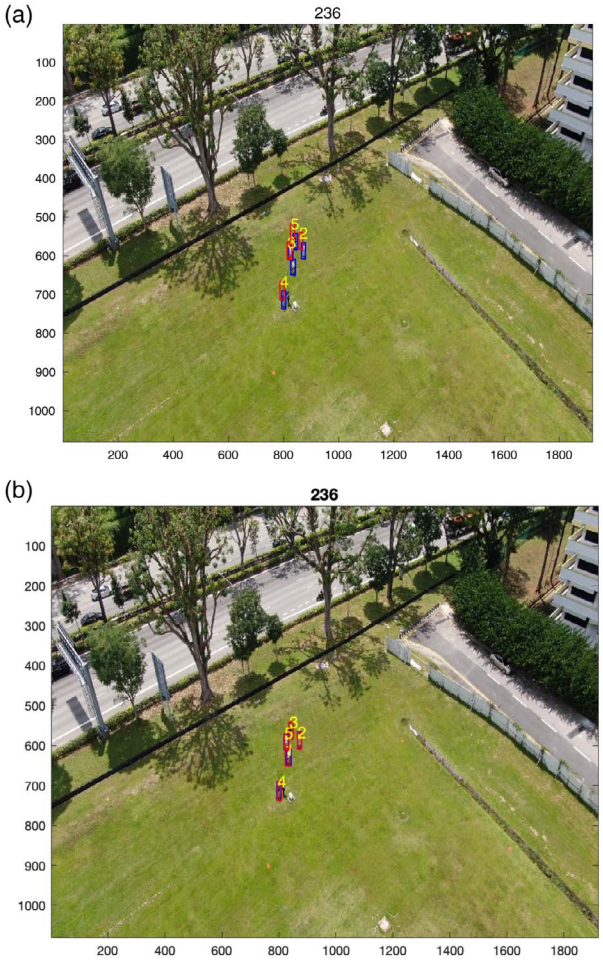


Figure 4. HTX scenario: tracking results at frame 236 based on (a) the GMIC method and (b) the proposed method (MLE4). Blue bounding boxes are the actual measurements, and the red bounding boxes are the tracking results. (a) Tracking based on the GMIC method. (b) Tracking based on MLE4.

the measured position based on the most recent assignment) of each target.

The position residual errors of the GMIC (gating) versus MLE3 are illustrated in Fig. 3. The camera adjusts its FoV to track targets, resulting in camera motion changes around frames 70 and 150–300. Our proposed MLE3 (assignment with exhaustive search) yields much smaller position residual errors than GMIC without camera motion parameters estimation. The ATRE for GMIC and MLE3, MLE4 are 9.04, 2.94, and 2.48, respectively. MLE4 has ATRE about 73% smaller than the GMIC method. The results demonstrate the robustness of the proposed methods. MLE3 and MLE4 have the ability to maintain tracking accuracy even in dynamic camera scenarios, while GMIC's reliance on validation gates can lead to incorrect association and degraded tracking in the presence of sudden camera movements.

Figure 4 shows the tracking results (GMIC versus MLE4) at frame 236. The blue BBs represent actual measurements, and red BBs indicate tracking results. For

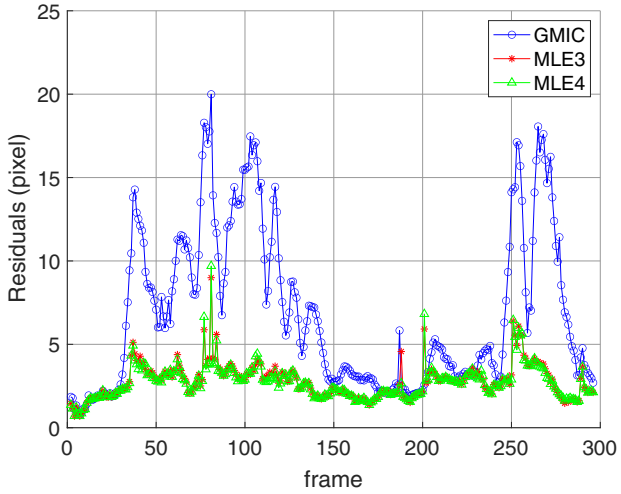


Figure 5. VDD scenario 1: Average track residual errors for GMIC, MLE3, and MLE4 are 6.93, 2.77, and 2.72, respectively.

MLE4, the red BBs for all targets almost cover the blue BBs, whereas for GMIC, the red BBs have large deviations from the blue BBs. This indicates that the proposed MLE4 performs better tracking than the GMIC when the camera has motion change at frame 236.

C. Large Number of Targets—VDD Video, Assignment With Hungarian

Two scenarios with multiple targets from VisDrone Dataset (VDD)[30][9] are considered. The sampling frequency is 15 Hz (15 frames/s), and the frame has a size of 1344 $\times$ 756 px. The UAS can change its position and altitude and thus result in more complex camera motion change.

For Scenario 1 (VDD #0000088_00290), there are more than 60 targets in one frame. The average size (width and height) of the BBs is around [35 80] px. The average velocity of the BBs is around [5 45] px/s. The process noise power spectral density is chosen as[10] $q = 49$ (px^2/s^3) and the measurement noise covariance matrix as $R = \text{diag}[9\ 9]$ (px^2).

As shown in Fig. 5, the position residual errors of the GMIC method are significantly larger compared to those obtained with MLE3 method and MLE4 method (assignment with Hungarian), especially when the camera is panning or tilting during frames 80–140 and 250–270. The average track residual errors during the whole frames for GMIC, MLE3 and MLE4 are 6.93, 2.77, and 2.72, respectively. The tracking results (GMIC versus MLE4) at frame 105 are shown in Fig. 6.

For Scenario 2 (VDD #0000099_02109), the average size (width and height) of the BBs is around [20 50] px. The average velocity of the BBs is around [4 40] px/s.

[9]https://github.com/VisDrone/VisDrone-Dataset.

[10]The RMS change in the velocity over a sampling interval T is $\sqrt{qT}$, which for $T = 1/15$ s, yields 1.8 px/s.

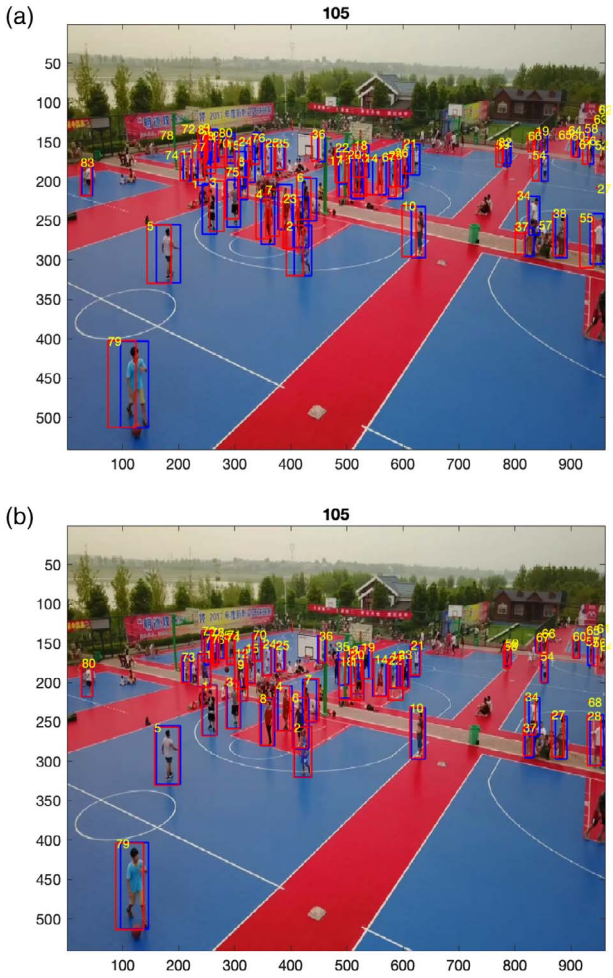


Figure 6. VDD scenario 1: tracking results at frame 105 based on (a) the GMIC method and (b) the proposed method (MLE4). (a) Tracking based on the GMIC method. (b) Tracking based on MLE4.

Thus, we choose the process noise power spectral density as[11] $q = 16$ (px^2/s^3) and the measurement noise covariance matrix as $R = \text{diag}[9\ 9]$ (px^2).

Similarly, our proposed methods (MLE3 and MLE4) show smaller position residual errors than the GMIC method (ATRE = 6.47, 2.31, 2.12), as seen in Fig. 7, especially around frame 600. Note that the MLE4 outperforms MLE3. This is because the camera increases its altitude during frames 500–650 and has slight roll. The tracking results at frame 570 are shown in Fig. 8, revealing that MLE4 significantly surpasses GMIC in tracking accuracy.

Next, we evaluate in detail the tracking quality for Scenario 2. There are two metrics to evaluate the tracking performance: the number of track breaks and the number of track swaps. As shown in Table II, the proposed MLE methods outperform the GMIC method, yielding fewer track breaks and track swaps. Fig. 9 shows the track swaps and track breaks when the camera has

[11]This corresponds to an RMS change in the velocity over $T = 1/15$ s of 1 px/s.

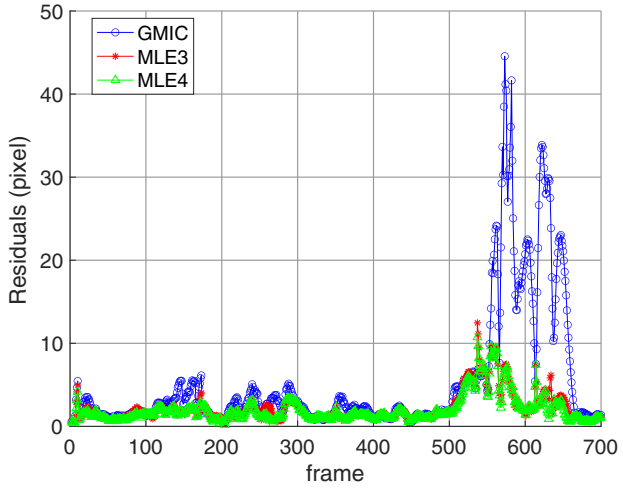


Figure 7. VDD scenario 2: Average track residuals errors for GMIC, MLE3, and MLE4 are 6.47, 2.31, and 2.12, respectively.

abrupt movements. Track breaks refer to interruptions in the continuous tracking of a target. Track swaps refer to the instances where the identity of a track is mistakenly switched between two different targets. The top three figures show that there are track IDs mistakenly swapped by the GMIC method, whereas the bottom figures show no track swaps by the MLE4. Due to the obstruction of buildings (yellow pavilion), track 13 from the top figures is broken, while MLE4 continues track 13 without track breakage or swap.

VI. CONCLUSIONS

In this paper, we carried out M2TA with camera motion parameters estimation for drone-captured video to reduce the effect of sudden camera movement. The camera parameter vector is extended from 3D (pan, tilt, and zoom) to 4D (pan, tilt, zoom, and roll). Based on the estimation, the proposed approach not only yields improved M2TA pairings but also can provide better state estimation in the update step. The real data results show that the proposed approach (MLE4 and MLE3) can reduce the effects of sudden camera movement. MLE4 is better than MLE3. The robustness has been confirmed by using the algorithm in diverse situations with good results. The tracking accuracy and tracking quality are much better than the GMIC (gating) method.

Table II
Tracking Quality for Scenario 2

	# Track Breaks	# Track Swaps
GMIC	5	5
MLE3	3	1
MLE4	2	1

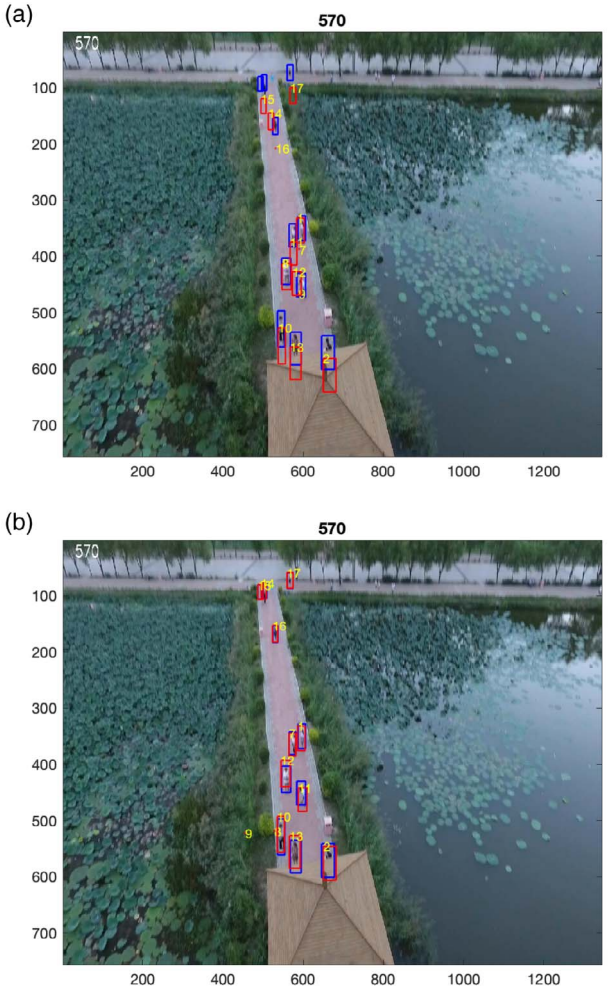


Figure 8. VDD scenario 2: tracking results at frame 570 based on (a) the GMIC method and (b) the proposed method (MLE4). (a) Tracking based on the GMIC method. (b) Tracking based on MLE4.

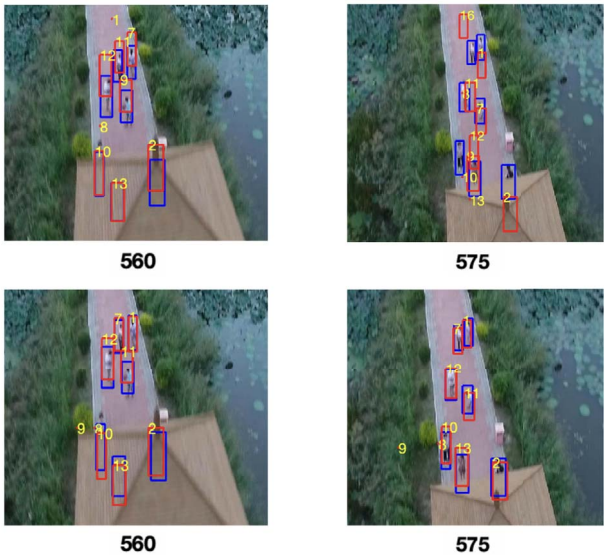


Figure 9. Track breaks and track swaps. Top figures: Tracks 9, 12, and 13 are mistakenly swapped by the GMIC method from frame 560 to frame 575 when the camera has abrupt movements. Bottom figures: There are no track swaps by the MLE4.

APPENDIX A

The LLS method is used to solve for $\hat{\boldsymbol{\zeta}}$ directly in the MLE3 method. For an assignment with $\{i \leftrightarrow j\}_{i=1}^{N}$ (N is the number of M2TA pairings), the cost based on (4) is expressed as

$$c[i,\, j(i),\, k,\, \boldsymbol{\zeta}(k)] = ||\mathbf{z}_i(k) - \hat{\mathbf{z}}^{\kappa}_{j(i)}(k|k-1)||^2 \tag{35}$$

$$= [x_i(k) - \hat{x}_{j(i)}(k|k-1)\phi(k) - x_c(k)]^2 + [y_i(k) - \hat{y}_{j(i)}(k|k-1)\phi(k) - y_c(k)]^2 \tag{36}$$

where $j(i)$ denotes the track paired with measurement i.[12] Define the following stacked matrices:

$$\mathbf{H} = [\mathbf{H}_1 \ldots \mathbf{H}_N]', \tag{37}$$

$$\mathbf{v} = [\mathbf{v}_1 \ldots \mathbf{v}_N]', \tag{38}$$

$$\mathbf{R} = \text{diag}[\Re_1 \ldots \Re_N], \tag{39}$$

where

$$\mathbf{H}_j = \begin{bmatrix} \hat{x}_j & 1 & 0 \\ \hat{y}_j & 0 & 1 \end{bmatrix}, \tag{40}$$

$$\mathbf{v}_j = \begin{bmatrix} x_{i(j)} - \hat{x}_j \\ y_{i(j)} - \hat{y}_j \end{bmatrix}, \tag{41}$$

$$\Re_j = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}, \tag{42}$$

with the estimate of the innovation variance (for statistical significance, to be discussed later) is

$$\hat{\sigma}^2 = \frac{1}{2N - n_\zeta}(\mathbf{H}\hat{\boldsymbol{\zeta}} - \mathbf{v})'(\mathbf{H}\hat{\boldsymbol{\zeta}} - \mathbf{v}), \tag{43}$$

where $i(j)$ in (41) is the index of the measurement associated with track j, $j = 1, \ldots, N$ and n_ζ is the number of camera parameters.

Then the LLS problem for pairs $i(j)$, j is given by

$$\arg\min_{\zeta} \sum_{j=1}^{N} (\mathbf{H}_j\boldsymbol{\zeta} - \mathbf{v}_j)'\Re_j^{-1}(\mathbf{H}_j\boldsymbol{\zeta} - \mathbf{v}_j), \tag{44}$$

or, without the summation (with the stacked matrices)

$$\arg\min_{\zeta}(\mathbf{H}\boldsymbol{\zeta} - \mathbf{v})'\mathbf{R}^{-1}(\mathbf{H}\boldsymbol{\zeta} - \mathbf{v}), \tag{45}$$

Finally, $\hat{\boldsymbol{\zeta}}$ is obtained by minimizing the quadratic error (44),

$$\hat{\boldsymbol{\zeta}} = (\mathbf{H}'\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}'\mathbf{R}^{-1}\mathbf{v} = \left(\sum_{j=1}^{N}\mathbf{H}_j'\mathbf{H}_j\right)^{-1}\mathbf{H}'\mathbf{v}, \tag{46}$$

with its covariance matrix given by

$$\mathbf{P}_{\hat{\zeta}} = (\mathbf{H}'\mathbf{R}^{-1}\mathbf{H})^{-1} = \left(\sum_{j=1}^{N}\mathbf{H}_j'\Re_j^{-1}\mathbf{H}_j\right)^{-1}. \tag{47}$$

[12] The pairing notations $i(j)$ and $j(i)$ are equivalent.

APPENDIX B

The partial derivatives equation (20) in Section III.B are as follows:

$$\frac{\partial h_1}{\partial \rho} = (-\hat{x}_{i(j)}(k|k-1)\sin\rho(k) + \hat{y}_{i(j)}(k|k-1)\cos\rho(k))\phi(k), \tag{48}$$

$$\frac{\partial h_1}{\partial \phi} = \hat{x}_{i(j)}(k|k-1)\cos\rho(k) + \hat{y}_{i(j)}(k|k-1)\sin\rho(k), \tag{49}$$

$$\frac{\partial h_1}{\partial x_c} = 1, \tag{50}$$

$$\frac{\partial h_1}{\partial y_c} = 0, \tag{51}$$

$$\frac{\partial h_2}{\partial \rho} = (-\hat{y}_{i(j)}(k|k-1)\sin\rho(k) - \hat{x}_{i(j)}(k|k-1)\cos\rho(k))\phi(k), \tag{52}$$

$$\frac{\partial h_2}{\partial \phi} = (\hat{y}_{i(j)}(k|k-1)\cos\rho(k) - \hat{x}_{i(j)}(k|k-1)\sin\rho(k)), \tag{53}$$

$$\frac{\partial h_2}{\partial x_c} = 0, \tag{54}$$

$$\frac{\partial h_2}{\partial y_c} = 1. \tag{55}$$

REFERENCES

[1] G. Allebosch, D. V. Hamme, P. Veelaert, and W. Philips
"Robust pan/tilt compensation for foreground–background segmentation,"
Sensors, vol. 19, no. 12, p. 2668, 2019.

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

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

[4] S. Blackman and R. Popoli
Design and Analysis of Modern Tracking Systems, Dedham, MA, USA: Artech House, 1999.

[5] C. Cadena et al.
"Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,"
IEEE Trans. Robot., vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[6] A. Censi, A. Fusiello, and V. Roberto
"Image stabilization by feature tracking,"
Proc. 10th Int. Conf. Image Anal. Process., Venice, Italy, 1999, pp. 665–667.

[7] M.-N. Chapel and T. Bouwmans
"Moving objects detection with a moving camera: A comprehensive review,"
Comput. Sci. Rev., vol. 38, p. 100310, 2020.

[8] W. Choi, C. Pantofaru, and S. Savarese
"A general framework for tracking multiple people from a moving camera,"
IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 7, pp. 1577–1591, Jul. 2013.

[9] J. Edmonds and R. Karp
"Theoretical improvements in algorithmic efficiency for network flow problems,"
J. ACM. vol. 19, no. 2, pp. 248–264, Apr. 1972.

[10] S. Erturk
"Digital image stabilization with sub-image phase correlation based global motion estimation,"
Trans. Consum. Electron., vol. 49, no. 4, pp. 1320–1325, Nov. 2003.

[11] G. D. Evangelidis and E. Z. Psarakis
"Parametric image alignment using enhanced correlation coefficient maximization,"
IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 10, pp. 1858–1865, Oct. 2008.

[12] B. Kiefer et al.
"1st workshop on maritime computer vision (MaCVi) 2023: Challenge results," *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 265–302.

[13] S. Li and D.-Y. Yeung
"Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models,"
AAAI, vol. 31, no. 1, Feb. 2017.

[14] A. Litvin, J. Konrad, and W. Karl
"Probabilistic video stabilization using Kalman filtering and mosaicking,"
Proc. SPIE, vol. 5022, pp. 663–674, May 2003.

[15] E. Mingkhwan and W. Khawsuk
"Digital image stabilization technique for fixed camera on small size drone,"
in *Proc. 3rd Asian Conf. Defence Technol.* 2017, pp. 12–19.

[16] M. Mueller, G. Sharma, N. Smith, and B. Ghanem
"Persistent aerial tracking system for UAVs," 2016
IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Daejeon, Korea (South), 2016, pp. 1562–1569.

[17] M. Mueller, N. Smith, and B. Ghanem
"A Benchmark and Simulator for UAV Tracking,"
in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 445–461.

[18] A. Ramachandran and A. K. Sangaiah
"A review on object detection in unmanned aerial vehicle surveillance,"
Int. J. Cogn. Comput. Eng., vol. 2, pp. 215–228, 2021.

[19] P. Rawat and J. Singhai
"Review of motion estimation and video stabilization techniques for hand held mobile video," signal and image process:
An Int. J., vol. 2, no. 2, pp. 159–168, Jun. 2011.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi
"You only look once: Unified, real-time object detection,"
in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788.

[21] J. Redmon and A. Farhadi
"Yolov3: An incremental improvement," 2018,
arXiv:1804.02767.

[22] D. Reid
"An algorithm for tracking multiple targets,"
IEEE Trans. Autom. Control, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[23] W. Sohn and N. D. Kehtarnavaz
"Analysis of camera movement errors in vision-based vehicle tracking,"
IEEE Trans. Pattern Anal. Mach. Intell., vol. 17, no. 1, pp. 57–61, Jan. 1995.

[24] J. Thomas, J. Welde, G. Loianno, K. Daniilidis, and V. Kumar
"Autonomous flight for detection, localization, and tracking of moving targets with a small quadrotor,"
IEEE Robot. Automat. Lett., vol. 2, no. 3, pp. 1762–1769, Jul. 2017.

[25] Z. Tian, Y. Bar-Shalom, R. Yang, H. J. Huang, and G. W. Ng
"Interframe association of YOLO bounding boxes in the presence of camera panning and zooming,"
in *Proc. 26th Int. Conf. Inf. Fusion*, 2023, pp. 1–7.

[26] B.-N. Vo et al.
"Multitarget tracking,"
in *Wiley Encyclopedia of Electrical and Electronics Engineering*. New York, NY, USA: Wiley, 2015.

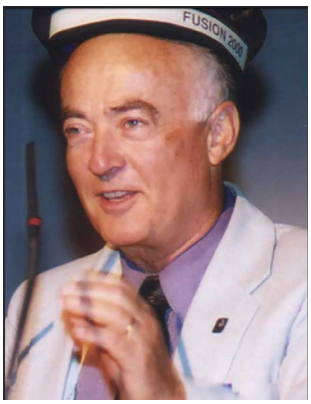
[27] S. Yeom and I.-J. Cho
"Detection and tracking of moving pedestrians with a small unmanned aerial vehicle," Appl. Sci., vol. 9, Art. no. 3359, 2019.

[28] J. Yang, D. Schonfeld, and M. Mohamed
"Robust video stabilization based on particle filter tracking of projected camera motion,"
IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 7, pp. 945–954, Jul. 2009.

[29] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu
"Object detection with deep learning: A review,"
IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[30] P. Zhu et al.
"Detection and tracking meet drones challenge,"
IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 11, pp. 7380–7399, Nov. 2022.

Zijiao Tian received her B.Sc. degree in electrical engineering from Xidian University, Xi'an, Shaanxi, China, in 2019, M.Sc. and Ph.D. degrees in electrical engineering from the University of Connecticut (UConn), Storrs, CT, USA, in 2023 and 2024, respectively. She is currently a Research Scientist with Intelligent Fusion Technology, Inc (IFT), Germantown, MD, USA. Her research interests include statistical signal processing, target detection and tracking, sensor fusion, positioning, navigation, and timing (PNT). She was awarded the Summer Doctoral Dissertation Fellowship at UConn in 2024.



Yaakov Bar-Shalom (IEEE F'84) received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion in 1963 and 1967, respectively, and the Ph.D. degree 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 over more than 650 papers and book chapters (more than 71 000 citations, $h = 103$). 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 Transactions on Automatic Control and Automatica. He was General Chairman of the 1985 ACC, General Chairman of FUSION 2000, President of ISIF in 2000 and 2002, and Vice President for Publications from 2004 to 2013. Since 1995, he has been a Distinguished Lecturer of the IEEE AESS. He is a corecipient of the M. Barry Carlton Award for the best paper in the IEEE TAE Systems 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 ISIF 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. He is also the recipient (with H. Blom) of the IEEE AESS Pioneer Award for the invention of the IMM Estimator. He has been listed in the Stanford Top 2% Researchers List in 2023. He is also co-recipient of the 2023 Naval Research Lab Alan Berman Research Publication Award.

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

Huang Hong'An Jack was born in Singapore in 1983. He received the B.E. degree from National University of Singapore (NUS), Singapore, in 2008. He is currently a Senior Member of Technical Staff with DSO National Laboratories, Singapore. His research interests include target tracking, including GMTI tracking, passive tracking, and image tracking. He received the FUSION 2014 Best Paper Award (First Runner-Up).

Gee Wah Ng received the M.Sc. and Ph.D. degrees from the University of Manchester Institute of Science and Technology, Manchester, UK. He is currently a Distinguished Member of Technical Staff at DSO National Laboratories, Singapore, and a Director at Home Team Science and Technology Agency (HTX), Singapore. He has delivered many projects in the decision support areas and has authored three books. He is active in international conferences in the areas of information fusion and intelligent systems. His research interests in data and information fusion include target tracking, computational intelligence, machine learning, self-tuning, and sensor networks. He was the Program Chair of FUSION 2012 and received the FUSION 2014 Best Paper Award (First Runner-Up).

INTERNATIONAL SOCIETY OF INFORMATION FUSION

ISIF Website: http://www.isif.org

2024 BOARD OF DIRECTORS*

2022–2024	2023–2025	2024–2026
Felix Govaers	Gustaf Hendeby	Paulo Costa
Lyudmila Mihaylova	Wolfgang Koch	Manon Kok
Paul Thomas	Claire Laudy	Terry Ogle

*Board of Directors are elected by the members of ISIF for a three year term.

PAST PRESIDENTS

Simon Maskell, 2022	Darin Dunham, 2014	Pierre Valin, 2006
Simon Maskell, 2021	Wolfgang Koch, 2013	W. Dale Blair, 2005
Paulo Costa, 2020	Roy Streit, 2012	Chee Chong, 2004
Paulo Costa, 2019	Joachim Biermann, 2011	Xiao-Rong Li, 2003
Lyudmila Mihaylova, 2018	Stefano Coraluppi, 2010	Yaakov Bar-Shalom, 2002
Lyudmila Mihaylova, 2017	Elisa Shahbazian, 2009	Pramod Varshney, 2001
Jean Dezert, 2016	Darko Musicki, 2008	Yaakov Bar-Shalom, 2000
Darin Dunham, 2015	Erik Blasch, 2007	Jim Llinas, 1999

SOCIETY VISION

The International Society of Information Fusion (ISIF) is the premier professional society and global information resource for multidisciplinary approaches for theoretical and applied information fusion technologies.

SOCIETY MISSION

Advocate

To advance the profession of fusion technologies, propose approaches for solving real-world problems, recognize emerging technologies, and foster the transfer of information.

Serve

To serve its members and engineering, business, and scientific communities by providing high-quality information, educational products, and services.

Communicate

To create international communication forums and hold international conferences in countries that provide for interaction of members of fusion communities with each other, with those in other disciplines, and with those in industry and academia.

Educate

To promote undergraduate and graduate education related to information fusion technologies at universities around the world. Sponsor educational courses and tutorials at conferences.

Integrate

Integrate ideas from various approaches for information fusion, and look for common threads and themes– look for overall principles, rather than a multitude of point solutions. Serve as the central focus for coordinating the activities of world-wide information fusion related societies or organizations. Serve as a professional liaison to industry, academia, and government.

Disseminate

To propagate the ideas for integrated approaches to information fusion so that others can build on them in both industry and academia.

Call for Papers

The Journal of Advances in Information Fusion (JAIF) seeks original contributions in the technical areas of research related to information fusion. Authors are encouraged to submit their manuscripts for peer review http://isif.org/journal.

Call for Reviewers

The success of JAIF and its value to the research community is strongly dependent on the quality of its peer review process. Researchers in the technical areas related to information fusion are encouraged to register as a reviewer for JAIF at http://jaif.msubmit.net. Potential reviewers should notify via email the appropriate editors of their offer to serve as a reviewer.

Journal of Advances in Information Fusion Volume 19 Number 1 June 2024 ISSN 1557-6418