June 2011 Volume 6 Number 1 ISSN 1557-6418

JAIF

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.

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

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

From the Editor-In-Chief

June 2011

Transforming a Conference Paper into a Journal Article

The ISIF VP for Conferences and the editorial board for JAIF regularly encourage authors to submit their papers that have been published in the Proceedings of the International Conference on Information Fusion (ICIF) to JAIF. Papers published in peer-reviewed journals such as JAIF receive much broader archival exposure, as I discussed in my editorial in the June 2008 issue. However, the expectations and standards for peer-reviewed journal papers are quite higher than those for conference papers. Over the years, I have authored numerous conference papers and after some maturing of the contributions in two or three of those papers, I have integrated the results into a journal submission. For example, I authored a paper in the 2008 ICIF on the design of nearly constant velocity track filters for sustained maneuvers. After the conference, I decided that the design procedures should be extended to brief maneuvers to make a contribution worthy a journal paper. I also noted that improvements were needed in the notation and presentation of the method. In the 2011 ICIF, I authored a paper that extended the results of 2008 to brief maneuvers and used the improved notation. Now, I am preparing a journal submission on the design of nearly constant velocity filters for tracking maneuvering targets.

Based on my experience, I have developed some suggestions for transforming a conference paper into a manuscript that will be successful in the peer review process of JAIF or other peer-reviewed journal.

1. **Significance of Contribution** One should first consider the significance of the contribution of the conference paper. Is the contribution novel? Will others build upon this research? Will others use it in their research or work? In five years, will the results still be important? If the results are novel and the answers to a couple of these questions are yes, then you should consider preparing a journal submission. An author of a conference paper is rarely left without an idea for improving the contribution in their paper. Then, make a list of potential improvements

that you have identified in the process of writing and presenting the paper. From the list of improvements, identify the ones that make the contribution whole without a disproportionately large level of effort and then complete those contributions.

2. **Introduction** Revise the introduction as needed to clearly describe the problem under consideration and include a very thorough survey of the related literature with references. Carefully and succinctly describe the contribution of the paper. Authors tend to be a little ambiguous in the description of the contribution of their paper, and this is a common mistake because referees are expected (and can be relied on) to doubt and challenge the contribution of a paper. Ambiguity in the statement of the contribution tends to raise concerns with the referees. Remember that it is better to have a small contribution that is well elucidated than a great contribution that is questionable.
3. **Problem Definition and Background Material** Review your formulation, notation and definition of the problem and make improvements based on your experience with the conference paper. Also, add or remove background material as appropriate.
4. **Contribution** Examine the core contributions of the paper for potential improvement in the presentation of those contributions. Look for missing items and potential extensions in the development of the contributions. If a referee cannot follow the development of the contributions, your manuscript is likely to be rejected.
5. **Example or Simulation Results** Consider improving the paper with a better example or different results. You should consider using a different example and pointing the reader to the example in your conference paper for additional results. Also, make sure that you have included all of the parameters and simulation details needed to reconstruct the results. (Otherwise, the only suitable place for submission is JIR—the Journal of Irreproducible Results.) Failure to give a proper level of details from the simulation results raises questions with the referees. The feedback that one receives at a conference most often addresses this topic. Compare your results with what you think is the best "competition" in the literature.
6. **Concluding Remark** In the concluding section (and there must be one!), concisely summarize the contributions of the paper and include any limitations of your research. If you do not provide the limitations of your results, the referee will likely make an assessment based on an application you do not want, and he/she will have less motivation to have the manuscript published in the JAIF. Also, address the significance of the contribution. This is the author's opportunity to make the case for publication of the manuscript. Then, the author should provide some comments on further research and these comments will support the significance of the paper.
7. **Abstract** After finishing your revision of the manuscript for journal submission, review the abstract to ensure that it accurately reflects the new version of the manuscript.

I encourage authors of ICIF papers to transform their conference papers for journal submissions for JAIF. Publication of your research in a peer-reviewed journal such as JAIF will give it much broader archival exposure. Furthermore, the peer-review process will certainly improve your research and leave you with the sense of accomplishment.

William Dale Blair
Editor In Chief

Tracking with Multisensor Out-of-Sequence Measurements with Residual Biases

SHUO ZHANG
YAAKOV BAR-SHALOM
GREGORY WATSON

In multisensor target tracking systems, measurements from different sensors on the same target typically exhibit biases. These biases can be accounted for as fixed random variables by the Schmidt-Kalman filter. Furthermore, measurements from the same target can arrive out of sequence. Recently, a procedure for updating the state with a multistep-lag "out-of-sequence" measurement (OOSM) using the simpler "1-step-lag" algorithm was developed for the situation without measurement biases. The present work presents the solution to the combined problem of handling biases from multiple sensors when their measurements arrive out of sequence. The state update with an OOSM is derived first for a KF tracker. This technique is then extended to the case where the tracker is an IMM estimator.

Manuscript received December 18, 2009; revised May 17, 2010 and November 1, 2010; released for publication November 17, 2010.

Refereeing of this contribution was handled by Stefano Coraluppi.

Research sponsored under ONR N00014-07-1-0131 and ARO W911NF-06-1-0467 grants.

Authors' addresses: S. Zhang and Y. Bar-Shalom, ECE Department, University of Connecticut, Storrs, CT 06269-2157, E-mail: (shuo.zhang@engr.uconn.edu); G. Watson, SPARTA, Arlington, VA 22209, E-mail: (Gregory.Watson@sparta.com).

1. INTRODUCTION

In multisensor target tracking systems, measurements from different sensors on the same target typically exhibit biases. These biases can be accounted for as fixed random variables by the Schmidt-Kalman filter. Furthermore, measurements from the same target can arrive out of sequence. Recently, a procedure for updating the state with a multistep-lag "out-of-sequence" measurement (OOSM) using the simpler "1-step-lag" algorithm was developed for the situation without measurement biases. The present work presents the solution to the combined problem of handling biases from multiple sensors when their measurements arrive out of sequence. The state update with an OOSM is derived first for a KF tracker. This technique is then extended to the case where the tracker is an IMM estimator.

The OOSM problem has been discussed in the literature starting with the initial work of [6] (discussed also in [5]), which presented an approximate solution to the problem of updating the current state of a target with an one-step-lag OOSM, called "algorithm B" in [2]. The optimal solution to the one-step-lag OOSM problem, called "algorithm A," was derived in [2]. It was also shown in [2] that algorithm B is nearly optimal for a one-step-lag OOSM. In [10], the comparison of algorithms A and B is discussed. In the case of receiving more that one OOSM in succession, one needs to modify algorithm A slightly (to preserve the optimality): in addition to updating the state at the current time, one also needs to update the state at the OOSM time using the standard Kalman updating algorithm. Alternatively, one can also stack multiple OOSMs in a single vector and use (the augmented version of) algorithm A to update the state with multiple OOSMs optimally in one step (see [17] for more details). In all these works it was assumed that the OOSM lag is less than a sampling interval. This has been designated as the "one-step-lag OOSM problem," and thus the corresponding algorithms can be called A1 and B1. The first solution to the general l-step-lag OOSM problem, Bl, was presented in [11] in the framework of B1. The algorithm Bl requires the storage of the sequence of filter gains and measurement matrices. The approach presented in [3] obtains the update with an l-step-lag OOSM in a *single step* (a "giant leap"), i.e., it generalized the previous algorithms to an arbitrary l. Furthermore, the resulting algorithms, Al1 and Bl1, have practically the same requirements as those of A1 and B1, respectively, for all $l > 1$. These algorithms have also been shown to perform nearly optimally in [3]. A general optimal solution to the OOSM problem was presented in [19], but it is substantially more complicated than [3].

A particle filter (PF) approach for dealing with OOSMs with arbitrary lags is proposed in [14], which presented a general solution to the nonlinear/non-Gaussian tracking problem in the presence of OOSMs. It was observed in [12] that, accuracywise, PF has no advantage over KF with converted Cartesian measurements or EKF, but takes much more CPU time. It was shown in [13] that the OOSM problem can be posed as a generalized smoothing or retrodiction problem and the Rauch-Tung-Streibel (RTS) smoother was used to obtain (in the linear case) an optimal algorithm for l-step-lag OOSM. Recently, a joint probability density approach called Accumulated State Density (ASD) is introduced in [8] with applications to the OOSM problem. By using ASD, the standard filtering and retrodiction are achieved in a unified manner. Rather than only updating the current state, ASD evaluates the effects of the OOSMs to all the states inside a certain window.

Section 2 presents the formulation of the OOSM problem for biased multiple sensors. Section 3 generalizes the Schmidt-Kalman filter (SKF), originally developed for tracking with a single sensor in the presence of residual biases, to the multisensor case. While the OOSM problem with biases from multiple sensors can be solved by augmenting the target state with all the sensor biases, this would not be practical for real systems. Section 4 derives the modified Joseph form for OOSM, which considers both the cases with and without biases. The combined problem of OOSM with biases from multiple sensors is solved in Section 5 using the Bl1 approach combined with the SKF *without state augmentation* resulting in the SKF/OOSM algorithm. These techniques are also described for the case where the tracker is an IMM estimator in Section 6. Section 7 discusses the heuristic "covariance inflation" approach for biases. The simulation results are given in Section 8. Section 9 presents a discussion of the results.

2. FORMULATION OF THE PROBLEM

The state of the system, x, of dimension n_x, is assumed to evolve from time t_{k-1} to time t_k according to

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

where, using only the index of the time arguments, $F(k,k-1)$ is the state transition matrix to time t_k from time t_{k-1} and $v(k,k-1)$ is the (cumulative effect of the) process noise for this interval. The order of the arguments in both F and v follows here the convention for the transition matrices. Typically, the process noise has a single argument, but here two arguments will be needed for clarity.

The measurement equation is

$$z^{i(k)}(k) = H_x^{i(k)}(k)x(k) + w^{i(k)}(k) + H_b^{i(k)}(k)b^{i(k)}, \qquad i \in \{1,\ldots,N_S\} \tag{2}$$

where $i(k)$ is the index of the sensor which provided the measurement[1] from time t_k (the "time stamp"), $w^{i(k)}(k)$ is the corresponding measurement noise, modelled to be zero-mean, and $b^{i(k)}$ is the residual bias for this sensor. The dimension of the above measurement is n_{z^i} and the dimension of the bias in this measurement is denoted as n_i. The matrix H_b multiplying the bias has been discussed in [15] for various nonlinear measurements.

It is assumed that bias correction has been done separately (externally to the OOSM problem) following a sensor registration procedure. Consequently, the residual bias, assumed to be a *time-invariant random variable*, is zero-mean

$$E[b^i] = 0, \qquad i \in \{1,\ldots,N_S\} \tag{3}$$

and

$$\mathrm{cov}[b^i,b^j] = E[b^i(b^j)'] = P_{b^ib^j}\delta_{ij}, \qquad i,j \in \{1,\ldots,N_S\} \tag{4}$$

where the shorter superscripts are used.

The noises are assumed zero-mean, white with covariances

$$\begin{aligned} E[v(k,j)v(k,j)'] &= Q(k,j) \\ E[w^{i(k)}(k)w^{i(k)}(k)'] &= R^{i(k)}(k) \end{aligned} \tag{5}$$

and, together with initial state error and the residual biases, mutually uncorrelated.

The time τ, at which the OOSM was made, is assumed to be such that

$$t_{k-l} < \tau < t_{k-l+1}. \tag{6}$$

This will require the evaluation of the effect of the process noise over an arbitrary noninteger number of sampling intervals. Note that $l = 1$ corresponds to the case where the lag is a fraction of a sampling interval; for simplicity this is called the "1-step-lag" problem, even though the lag is really a fraction of a time step.

The relationship between the current state $x(k)$ and the state observed by the OOSM is as follows. Similarly to (1), one has

$$x(k) = F(k,\kappa)x(\kappa) + v(k,\kappa) \tag{7}$$

where κ is the discrete time notation for τ. The above can be rewritten backward as

$$x(\kappa) = F(\kappa,k)[x(k) - v(k,\kappa)]. \tag{8}$$

where $F(\kappa,k) = F(k,\kappa)^{-1}$ is the backward transition matrix.

The problem is as follows: At time $t = t_k$ one has

$$\begin{aligned} \hat{x}(k \mid k) &\stackrel{\Delta}{=} E[x(k) \mid Z^k] \\ P(k \mid k) &\stackrel{\Delta}{=} \mathrm{cov}[x(k) \mid Z^k] \end{aligned} \tag{9}$$

based on the (multisensor) cumulative set of measurements at t_k

$$Z^k \stackrel{\Delta}{=} \{z^{i(\ell)}(\ell)\}_{\ell=1}^k. \tag{10}$$

[1] The superscript $i(k)$ will be shortened to i wherever this does not cause confusion.

Subsequently, the earlier measurement from time τ, denoted from now on with discrete time notation as κ,

$$z^{i(\kappa)}(\kappa) \stackrel{\Delta}{=} z^{i(\kappa)}(\kappa) = H_x^{i(\kappa)}(\kappa)x(\kappa) + w^{i(\kappa)}(\kappa) + H_b^{i(\kappa)}(\kappa)b^{i(\kappa)} \tag{11}$$

arrives after the state estimate (9) has been calculated. We want to update this estimate with the earlier measurement (11), namely, to calculate

$$\begin{aligned} \hat{x}(k \mid \kappa) &= E[x(k) \mid Z^\kappa] \\ P(k \mid \kappa) &= \mathrm{cov}[x(k) \mid Z^\kappa] \end{aligned} \tag{12}$$

where

$$Z^\kappa \stackrel{\Delta}{=} \{Z^k, z^{i(\kappa)}(\kappa)\}. \tag{13}$$

This update should be done *without* reordering and reprocessing the measurements according to their time stamps.

3. THE MULTISENSOR SCHMIDT-KALMAN FILTER

This section presents the multisensor Schmidt-Kalman Filter (SKF) for the case of state estimation in the presence of residual biases but without OOSMs. The SKF procedure [16, 7] consists of augmenting the target state vector with the measurement bias vector, calculating the KF gain for this augmented state but then updating only the target state. While the bias is not updated, its covariance stays constant, but the crosscovariance between the bias and the state does change when the state is updated.

In the multisensor case there are, however, as many bias vectors as the number of sensors from which measurements are obtained. Consequently, the straightforward approach would be to augment the target state with all the biases and, while only the target state is updated, the entire updated covariance matrix of such an augmented state has to be calculated, yielding all the updated state-bias crosscovariances. This approach can be, however, very costly because of the possibly high dimension of the augmented state—typically 6 for the target state and with a minimum of 3 bias components from possibly as many as 10 sensors (not an unlikely scenario), one has at least a 36×36-dimensional covariance matrix to be updated. The major problem with this high-dimensional matrix occurs in the update with the OOSM, which requires the inversion of the augmented state covariance matrix (which is a full matrix), and this can be computationally expensive for real time implementation.

In the development below it is shown that one can augment the target state only with the bias of the sensor which provided the measurement to be used for the update and a "generic" other sensor. This allows to obtain the updated crosscovariances of the state with all the biases, block by block, rather than having to update the covariance matrix of the state augmented with all the biases. A similar procedure will be used in the update with the OOSM to avoid the need to invert a very large matrix. Furthermore, in the OOSM case, the inversion will have to be done only for the $(n_x \times n_x)$ state covariance matrix, without any augmentation.

Let the augmented state, of dimension $n_x + n_i + n_j$, be

$$\mathbf{x} \stackrel{\Delta}{=} \begin{bmatrix} x \\ b^i \\ \beta^j \end{bmatrix} \tag{14}$$

where i is the index of the sensor that provided the measurement to be used for the update at time k (the time argument of this index is now dropped for simplicity) and j is the index of a "generic" other sensor. The "generic" sensor bias β^j includes all the sensor biases except that from the current measurement, e.g.,

$$\beta^j = \begin{bmatrix} b^2 \\ b^3 \end{bmatrix}, \qquad b^i = b^1 \tag{15}$$

for the case $N_S = 3$ and the current measurement is from sensor 1. The use of a single notation β^j is just for simplicity. The state equation for this augmented state is

$$\mathbf{x}(k) = \mathbf{F}(k,k-1)\mathbf{x}(k-1) + \mathbf{v}(k,k-1) \tag{16}$$

where

$$\mathbf{F}(k,k-1) \stackrel{\Delta}{=} \begin{bmatrix} F(k,k-1) & 0 & 0 \\ 0 & I_{n_i} & 0 \\ 0 & 0 & I_{n_j} \end{bmatrix} \tag{17}$$

I_{n_i} denotes the $n_i \times n_i$ identity matrix and

$$\mathbf{v}(k,k-1) \stackrel{\Delta}{=} \begin{bmatrix} v(k,k-1) \\ 0 \\ 0 \end{bmatrix} \tag{18}$$

i.e., the biases are assumed constant between their (external) updates. The measurement at time k is

$$z^i(k) = \mathbf{H}^i(k)\mathbf{x}(k) + w^i(k) \tag{19}$$

where

$$\mathbf{H}^i(k) \stackrel{\Delta}{=} [H_x^i(k) \quad H_b^i(k) \quad 0]. \tag{20}$$

Let the prediction covariance of $\mathbf{x}(k)$ be

$$\begin{aligned} &\mathbf{P}(k \mid k-1) \\ &\quad \stackrel{\Delta}{=} \begin{bmatrix} P_{xx}(k \mid k-1) & P_{xb^i}(k \mid k-1) & P_{x\beta^j}(k \mid k-1) \\ P_{xb^i}(k \mid k-1)' & P_{b^ib^i}(k \mid k-1) & 0 \\ P_{x\beta^j}(k \mid k-1)' & 0 & P_{\beta^j\beta^j}(k \mid k-1) \end{bmatrix} \\ &\quad = \begin{bmatrix} P_{xx}(k \mid k-1) & P_{xb^i}(k \mid k-1) & P_{x\beta^j}(k \mid k-1) \\ P_{xb^i}(k \mid k-1)' & P_{b^ib^i} & 0 \\ P_{x\beta^j}(k \mid k-1)' & 0 & P_{\beta^j\beta^j} \end{bmatrix}. \end{aligned} \tag{21}$$

Then the optimal filter gain for updating $\mathbf{x}(k)$ is

$$\mathbf{W}^i(k)^{\mathrm{OPT}} = \mathbf{P}(k \mid k-1)\mathbf{H}^i(k)' S^i(k)^{-1}$$
$$= \mathbf{P}(k \mid k-1)\mathbf{H}^i(k)' \cdot [\mathbf{H}^i(k)\mathbf{P}(k \mid k-1)\mathbf{H}^i(k)' + R^i(k)]^{-1}$$
$$= \begin{bmatrix} W_x^i(k) \\ W_{b^i}^i(k) \\ W_{\beta^j}^i(k) \end{bmatrix} \tag{22}$$

which consists of three blocks.

The idea of the SKF is to use only the top block from the above, i.e., the actual gain will be

$$\mathbf{W}^i(k) = \begin{bmatrix} W_x^i(k) \\ 0 \\ 0 \end{bmatrix} \tag{23}$$

The expression of this block is

$$W_x^i(k) = [P_{xx}(k \mid k-1)H_x^i(k)' + P_{xb^i}(k \mid k-1)H_b^i(k)']S^i(k)^{-1} \tag{24}$$

where the innovation covariance is

$$S^i(k) = H_x^i(k)P_{xx}(k \mid k-1)H_x^i(k)' + H_x^i(k)P_{xb^i}(k \mid k-1)H_b^i(k)' + H_b^i(k)P_{b^i x}(k \mid k-1)H_x^i(k)' + H_b^i(k)P_{b^i b^i}H_b^i(k)' + R^i(k). \tag{25}$$

Since (23) is a suboptimal gain, the state covariance[2] update equation to be used in this case is the Joseph form (see, e.g., [1], Eq. (5.2.3-18)), which is the only one valid for an arbitrary gain. Thus, we have

$$\mathbf{P}(k \mid k) = [I_{n_x+n_i+n_j} - \mathbf{W}^i(k)\mathbf{H}^i(k)]\mathbf{P}(k \mid k-1) \cdot [I_{n_x+n_i+n_j} - \mathbf{W}^i(k)\mathbf{H}^i(k)]' + \mathbf{W}^i(k)R^i(k)\mathbf{W}^i(k)'. \tag{26}$$

Using (20), (21), and (23), the blocks of (26) are obtained as

$$\begin{aligned} P_{xx}(k \mid k) &= [I_{n_x} - W_x^i(k)H_x^i(k)]P_{xx}(k \mid k-1)[I_{n_x} - W_x^i(k)H_x^i(k)]' \\ &\quad - W_x^i(k)H_b^i(k)P_{xb^i}(k \mid k-1)'[I_{n_x} - W_x^i(k)H_x^i(k)]' \\ &\quad - [I_{n_x} - W_x^i(k)H_x^i(k)]P_{xb^i}(k \mid k-1)H_b^i(k)'W_x^i(k)' \\ &\quad + W_x^i(k)H_b^i(k)P_{b^i b^i}H_b^i(k)'W_x^i(k)' + W_x^i(k)R^i(k)W_x^i(k)' \end{aligned} \tag{27}$$

$$P_{xb^i}(k \mid k) = [I_{n_x} - W_x^i(k)H_x^i(k)]P_{xb^i}(k \mid k-1) - W_x^i(k)H_b^i(k)P_{b^i b^i} \tag{28}$$

$$P_{x\beta^j}(k \mid k) = [I_{n_x} - W_x^i(k)H_x^i(k)]P_{x\beta^j}(k \mid k-1), \qquad \forall j \neq i(k) \tag{29}$$

$$P_{b^i b^i}(k) = P_{b^i b^i}(k-1) = P_{b^i b^i} \tag{30}$$

$$P_{\beta^j \beta^j}(k) = P_{\beta^j \beta^j}(k-1) = P_{\beta^j \beta^j} \tag{31}$$

$$P_{b^i \beta^j}(k) = 0. \tag{32}$$

The state update is done, in view of (23), according to

$$\hat{x}(k \mid k) = \hat{x}(k \mid k-1) + W_x^i(k)\nu^i(k) \tag{33}$$

where the innovation corresponding to $z^i(k)$ is

$$\nu^i(k) = z^i(k) - H_x^i(k)\hat{x}(k \mid k-1). \tag{34}$$

The prediction equations, based on the model (16) are the standard ones, namely,

$$\hat{x}(k \mid k-1) = F(k,k-1)\hat{x}(k-1 \mid k-1) \tag{35}$$

and for the covariance

$$\mathbf{P}(k \mid k-1) = \mathbf{F}(k,k-1)\mathbf{P}(k-1 \mid k-1)\mathbf{F}(k,k-1)' + \mathbf{Q}(k,k-1). \tag{36}$$

where

$$\mathbf{Q}(k,k-1) \stackrel{\Delta}{=} \mathrm{diag}[Q(k,k-1), 0_{n_i}, 0_{n_j}]. \tag{37}$$

The blocks of the prediction covariance (36) are calculated as

$$P_{xx}(k \mid k-1) = F(k,k-1)P_{xx}(k-1 \mid k-1)F(k,k-1)' + Q(k,k-1) \tag{38}$$

$$P_{xb^i}(k \mid k-1) = F(k,k-1)P_{xb^i}(k-1 \mid k-1) \tag{39}$$

$$P_{x\beta^j}(k \mid k-1) = F(k,k-1)P_{x\beta^j}(k-1 \mid k-1), \qquad \forall j \neq i(k). \tag{40}$$

Equations (28) and (29) yield the updated crosscovariances of the state with the bias in the measurement used in the update and with each bias in the other measurements, respectively. This procedure avoids having to handle the update of a potentially very large covariance matrix. The crosscovariance of the state with the bias in another sensor's measurement will be needed when that sensor's measurement becomes available for updating the state. Similarly, the predicted crosscovari-

[2] Actually this is not "state covariance" but "state-error covariance," since the state estimate is not the conditional mean any more due to the use of the suboptimal gain. However, for simplicity we still use the term "state covariance."

ances are obtained using (39) and (40) and they are the same for all the biases.

Thus, the above equations show how one can obtain the state estimate of the target accounting for all the biases in a multisensor situation, without resorting to state augmentation as far as the computations are concerned. The augmentation was used only to obtain the covariance matrix block updates.

4. MODIFIED JOSEPH FORM FOR OOSM

As discussed above, since the filter gain in the SKF is not optimal, the Joseph form should be used for covariance update. For an out-of-sequence measurement (OOSM), the time of the OOSM is not at the current time, so the Joseph should be modified accordingly. First, we consider the standard Joseph form. Then, the modified Joseph forms for OOSM are derived for the cases with and without residual biases.

4.1. Standard Joseph Form

The state model and measurement model are given by

$$x(k) = F(k)x(k-1) + v(k-1) \quad (41)$$

$$z(k) = H(k)x(k) + w(k) \quad (42)$$

where $v(k)$ and $w(k)$ are mutually independent white noise sequences with covariance $Q(k)$ and $R(k)$, respectively. The state estimate using a Kalman Filter is given by

$$\hat{x}(k \mid k) = \hat{x}(k \mid k-1) + W(k)\nu(k) \quad (43)$$

where W is the filter gain and ν is the innovation, which is given by

$$\begin{aligned}\nu(k) &= z(k) - H(k)\hat{x}(k \mid k-1)\\ &= H(k)x(k) + w(k) - H(k)\hat{x}(k \mid k-1). \quad (44)\end{aligned}$$

By substituting (44) into (43), the state estimate can be written as

$$\begin{aligned}\hat{x}(k \mid k) = \hat{x}(k \mid k-1) &+ W(k)H(k)[x(k) - \hat{x}(k \mid k-1)]\\ &+ W(k)w(k). \quad (45)\end{aligned}$$

Then, using (45) the estimation error at time k is given by

$$\begin{aligned}\tilde{x}(k \mid k) &= x(k) - \hat{x}(k \mid k)\\ &= [I - W(k)H(k)][x(k) - \hat{x}(k \mid k-1)] - W(k)w(k)\\ &= [I - W(k)H(k)]\tilde{x}(k \mid k-1) - W(k)w(k) \quad (46)\end{aligned}$$

where $\tilde{x}(k \mid k-1)$ is the prediction error. Thus, the error covariance $P(k \mid k)$ can be obtained as

$$\begin{aligned}P(k \mid k) &= \mathrm{cov}\{\tilde{x}(k \mid k)\}\\ &= [I - W(k)H(k)]\mathrm{cov}\{\tilde{x}(k \mid k-1)\}\\ &\quad \cdot [I - W(k)H(k)]' + W(k)\mathrm{cov}\{w(k)\}W(k)'\\ &= [I - W(k)H(k)]P(k \mid k-1)[I - W(k)H(k)]'\\ &\quad + W(k)R(k)W(k)' \quad (47)\end{aligned}$$

due to the fact that the prediction error $\tilde{x}(k \mid k-1)$ is independent of the measurement noise $w(k)$. Formula (47) is known as the Joseph form.

4.2. Modified Joseph Form For OOSM Without Residual Biases

Now, we consider an OOSM $z(\kappa)$ ($\kappa < k$). The most recent state estimate after receiving $z(\kappa)$ is given by [3]

$$\hat{x}(k \mid \kappa) = \hat{x}(k \mid k) + W(k,\kappa)\nu(\kappa) \quad (48)$$

where $\nu(\kappa)$ is the innovation at time κ of the OOSM, that is

$$\nu(\kappa) = z(\kappa) - H(\kappa)\hat{x}(\kappa \mid k). \quad (49)$$

Using the suboptimal technique B [5] (performed within 1% of the optimum when the OOSM has a one-step lag), the state retrodiction $\hat{x}(\kappa \mid k)$ is given by

$$\hat{x}(\kappa \mid k) = F(\kappa,k)\hat{x}(k \mid k) \quad (50)$$

and $\nu(\kappa)$ is obtained as

$$\begin{aligned}\nu(\kappa) &= z(\kappa) - H(\kappa)F(\kappa,k)\hat{x}(k \mid k)\\ &= H(\kappa)x(\kappa) + w(\kappa) - H(\kappa)F(\kappa,k)\hat{x}(k \mid k)\\ &= H(\kappa)F(\kappa,k)[x(k) - v(k,\kappa)]\\ &\quad + w(\kappa) - H(\kappa)F(\kappa,k)\hat{x}(k \mid k)\\ &= H(\kappa)F(\kappa,k)[x(k) - \hat{x}(k \mid k)]\\ &\quad - H(\kappa)F(\kappa,k)v(k,\kappa) + w(\kappa) \quad (51)\end{aligned}$$

which has made use of (8). Substituting (51) into (48), we have

$$\begin{aligned}\hat{x}(k \mid \kappa) = \hat{x}(k \mid k) &+ W(k,\kappa)H(\kappa)F(\kappa,k)[x(k) - \hat{x}(k \mid k)]\\ &- W(k,\kappa)H(\kappa)F(\kappa,k)v(k,\kappa) + W(k,\kappa)w(\kappa). \quad (52)\end{aligned}$$

Thus, the estimation error is

$$\begin{aligned}\tilde{x}(k \mid \kappa) &= x(k) - \hat{x}(k \mid \kappa)\\ &= [I - W(k,\kappa)H(\kappa)F(\kappa,k)][x(k) - \hat{x}(k \mid k)]\\ &\quad + W(k,\kappa)H(\kappa)F(\kappa,k)v(k,\kappa) - W(k,\kappa)w(\kappa)\\ &= [I - W(k,\kappa)H(\kappa)F(\kappa,k)]\tilde{x}(k \mid k)\\ &\quad + W(k,\kappa)H(\kappa)F(\kappa,k)v(k,\kappa) - W(k,\kappa)w(\kappa). \quad (53)\end{aligned}$$

Using (53), the error covariance is given by

$$
\begin{aligned}
P(k \mid \kappa) &= \operatorname{cov}\{\tilde{x}(k \mid \kappa)\} \\
&= [I - W(k,\kappa)H(\kappa)F(\kappa,k)]\operatorname{cov}\{\tilde{x}(k \mid k)\} \\
&\quad \cdot [I - W(k,\kappa)H(\kappa)F(\kappa,k)]' \\
&\quad + W(k,\kappa)H(\kappa)F(\kappa,k)\operatorname{cov}\{v(k,\kappa)\} \\
&\quad \cdot [W(k,\kappa)H(\kappa)F(\kappa,k)]' \\
&\quad + W(k,\kappa)\operatorname{cov}\{w(\kappa)\}W(k,\kappa)' \\
&\quad + [I - W(k,\kappa)H(\kappa)F(\kappa,k)] \\
&\quad \cdot \operatorname{cov}\{\tilde{x}(k \mid k), v(k,\kappa)\} \\
&\quad \cdot [W(k,\kappa)H(\kappa)F(\kappa,k)]' \\
&\quad + W(k,\kappa)H(\kappa)F(\kappa,k)\operatorname{cov}\{v(k,\kappa),\tilde{x}(k \mid k)\} \\
&\quad \cdot [I - W(k,\kappa)H(\kappa)F(\kappa,k)]' \\
&= [I - W(k,\kappa)H(\kappa)F(\kappa,k)]P(k \mid k) \\
&\quad \cdot [I - W(k,\kappa)H(\kappa)F(\kappa,k)]' \\
&\quad + W(k,\kappa)H(\kappa)F(\kappa,k)Q(k,\kappa) \\
&\quad \cdot [W(k,\kappa)H(\kappa)F(\kappa,k)]' \\
&\quad + W(k,\kappa)R(\kappa)W(k,\kappa)' \\
&\quad + [I - W(k,\kappa)H(\kappa)F(\kappa,k)]P_{xv}(k,\kappa \mid k) \\
&\quad \cdot [W(k,\kappa)H(\kappa)F(\kappa,k)]' \\
&\quad + W(k,\kappa)H(\kappa)F(\kappa,k)P_{xv}(k,\kappa \mid k)' \\
&\quad \cdot [I - W(k,\kappa)H(\kappa)F(\kappa,k)]'
\end{aligned}
\tag{54}
$$

due to the fact that the measurement noise $w(\kappa)$ of the OOSM is independent of the estimation error $\tilde{x}(k \mid k)$ and the process noise $v(k,\kappa)$. Note that, we have (as in [2] Eq. (22))

$$
\begin{aligned}
P_{xv}(k,\kappa \mid k) &= \operatorname{cov}\{\tilde{x}(k \mid k), v(k,\kappa)\} \\
&= \operatorname{cov}\{x(k), v(k,\kappa) \mid Z^k\}
\end{aligned}
\tag{55}
$$

since the covariance is independent of the conditioning Z^k. Therefore, when OOSM is considered and the state estimation is given by the technique B, the Joseph form should be modified as in (54).

4.3. Modified Joseph Form For OOSM With Residual Biases

Next, we derive the Joseph form by considering both OOSM and residual biases of the sensors, i.e., (54) for the state augmented with the biases. Using (14), the state equation for this augmented state evolving from κ (the time of the OOSM) to the current time k is given by

$$
\mathbf{x}(k) = \mathbf{F}(k,\kappa)\mathbf{x}(\kappa) + \mathbf{v}(k,\kappa) \tag{56}
$$

where

$$
\mathbf{F}(k,\kappa) = \begin{bmatrix} F(k,\kappa) & 0 & 0 \\ 0 & I_{n_i} & 0 \\ 0 & 0 & I_{n_j} \end{bmatrix} \tag{57}
$$

and

$$
\mathbf{v}(k,\kappa) = \begin{bmatrix} v(k,\kappa) \\ 0 \\ 0 \end{bmatrix}. \tag{58}
$$

The corresponding covariance of $\mathbf{v}(k,\kappa)$ is

$$
\mathbf{Q}(k,\kappa) = \begin{bmatrix} Q(k,\kappa) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{59}
$$

The OOSM at time κ obtained from sensor i is

$$
z^i(\kappa) = \mathbf{H}^i(\kappa)\mathbf{x}(\kappa) + w^i(\kappa) \tag{60}
$$

where

$$
\mathbf{H}^i(\kappa) = [H_x^i(\kappa) \quad H_b^i(\kappa) \quad 0]. \tag{61}
$$

Let the updated covariance of $\mathbf{x}(k)$ be

$$
\mathbf{P}(k \mid k) = \begin{bmatrix} P_{xx}(k \mid k) & P_{xb^i}(k \mid k) & P_{x\beta^j}(k \mid k) \\ P_{xb^i}(k \mid k)' & P_{b^ib^i} & 0 \\ P_{x\beta^j}(k \mid k)' & 0 & P_{\beta^j\beta^j} \end{bmatrix} \tag{62}
$$

and the crosscovariance between $\mathbf{x}(k)$ and $\mathbf{v}(k,\kappa)$ be

$$
\mathbf{P}_{xv}(k,\kappa \mid k) = \begin{bmatrix} P_{xv}(k,\kappa \mid k) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{63}
$$

due to the independence between the sensor biases and process noise. The SKF gain using the OOSM $z(\kappa)$ at time k is

$$
\mathbf{W}^i(k,\kappa) = \begin{bmatrix} W_x^i(k,\kappa) \\ 0 \\ 0 \end{bmatrix}. \tag{64}
$$

Then, using the modified Joseph form given in (54), the covariance for the state augmented with residual biases can be written as

$$
\begin{aligned}
\mathbf{P}(k \mid \kappa) &= [I_{n_x+n_i+n_j} - \mathbf{W}^i(k,\kappa)\mathbf{H}^i(\kappa)\mathbf{F}(\kappa,k)]\mathbf{P}(k \mid k) \\
&\quad \cdot [I_{n_x+n_i+n_j} - \mathbf{W}^i(k,\kappa)\mathbf{H}^i(\kappa)\mathbf{F}(\kappa,k)]' \\
&\quad + \mathbf{W}^i(k,\kappa)\mathbf{H}^i(\kappa)\mathbf{F}(\kappa,k) \\
&\quad \cdot \mathbf{Q}(k,\kappa)[\mathbf{W}^i(k,\kappa)\mathbf{H}^i(\kappa)\mathbf{F}(\kappa,k)]' \\
&\quad + \mathbf{W}^i(k,\kappa)R^i(\kappa)\mathbf{W}^i(k,\kappa)' \\
&\quad + [I_{n_x+n_i+n_j} - \mathbf{W}^i(k,\kappa)\mathbf{H}^i(\kappa)\mathbf{F}(\kappa,k)]\mathbf{P}_{xv}(k,\kappa \mid k) \\
&\quad \cdot [\mathbf{W}^i(k,\kappa)\mathbf{H}^i(\kappa)\mathbf{F}(\kappa,k)]' \\
&\quad + \mathbf{W}^i(k,\kappa)\mathbf{H}^i(\kappa)\mathbf{F}(\kappa,k)\mathbf{P}_{xv}(k,\kappa \mid k)' \\
&\quad \cdot [I_{n_x+n_i+n_j} - \mathbf{W}^i(k,\kappa)\mathbf{H}^i(\kappa)\mathbf{F}(\kappa,k)]'.
\end{aligned}
\tag{65}
$$

Using (56)–(64), the blocks of (65) are obtained as

$$
\begin{aligned}
P_{xx}(k \mid \kappa) = {} & [I_{n_x} - W_x^i(k,\kappa)H_x^i(\kappa)F(\kappa,k)]P_{xx}(k \mid k) \\
& \cdot [I_{n_x} - W_x^i(k,\kappa)H_x^i(\kappa)F(\kappa,k)]' \\
& - W_x^i(k,\kappa)H_b^i(\kappa)P_{xb^i}(k \mid k)' \\
& \cdot [I_{n_x} - W_x^i(k,\kappa)H_x^i(\kappa)F(\kappa,k)]' \\
& - [I_{n_x} - W_x^i(k,\kappa)H_x^i(\kappa)F(\kappa,k)] \\
& \cdot P_{xb^i}(k \mid k)H_b^i(\kappa)'W_x^i(k,\kappa)' \\
& + W_x^i(k,\kappa)H_b^i(\kappa)P_{b^ib^i}H_b^i(\kappa)'W_x^i(k,\kappa)' \\
& + W_x^i(k,\kappa)R^i(\kappa)W_x^i(k,\kappa)' \\
& + W_x^i(k,\kappa)H_x^i(\kappa)F(\kappa,k)Q(k,\kappa) \\
& \cdot [W_x^i(k,\kappa)H_x^i(\kappa)F(\kappa,k)]' \\
& + [I_{n_x} - W_x^i(k,\kappa)H_x^i(\kappa)F(\kappa,k)]P_{xv}(k,\kappa \mid k) \\
& \cdot [W_x^i(k,\kappa)H_x^i(\kappa)F(\kappa,k)]' \\
& + W_x^i(k,\kappa)H_x^i(\kappa)F(\kappa,k)P_{xv}(k,\kappa \mid k)' \\
& \cdot [I_{n_x} - W_x^i(k,\kappa)H_x^i(\kappa)F(\kappa,k)]'.
\end{aligned} \tag{66}
$$

The crosscovariance of the state at k with the bias of the OOSM evolves as

$$
\begin{aligned}
P_{xb^i}(k \mid \kappa) = {} & [I_{n_x} - W_x^i(k,\kappa)H_x^i(\kappa)F(\kappa,k)]P_{xb^i}(k \mid k) \\
& - W_x^i(k,\kappa)H_b^i(\kappa)P_{b^ib^i}.
\end{aligned} \tag{67}
$$

The crosscovariance of the state at k with the other biases evolve as

$$
P_{x\beta^j}(k \mid \kappa) = [I_{n_x} - W_x^i(k,\kappa)H_x^i(\kappa)F(\kappa,k)]P_{x\beta^j}(k \mid k), \quad \forall j \neq i(k) \tag{68}
$$

and the bias covariances stay unchanged, as in (30)–(32).

5. ONE-STEP ALGORITHM FOR MULTISTEP-LAG OOSM FOR MULTIPLE SENSORS WITH BIASES—SKF/OOSM

Using the approach of [3], one can perform in one step the update with an l-step-lag OOSM. Suitable modifications will be made to account for the fact that the measurements are biased. Two procedures, designated as Al1 and Bl1, were presented in [3] for the situation without biases. Both procedures retrodict the current state to the time of the OOSM, calculate the covariance of the retrodicted state, the retrodicted measurement and its the covariance, the crosscovariance between the current state and the retrodicted measurement and, with these, one can perform the direct update of the current state with the OOSM.

These algorithms are based on the 1-step-lag OOSM algorithms, designated in [2] as A and B, respectively. The difference between algorithms A and B is in the retrodiction of the current state to the time of the OOSM:

A) uses the exact conditional mean, which turns out to be an affine function of the current state estimate, with a second term being a linear transformation of the latest innovation;

B) uses a linear function of the current state estimate, which is the first term from the above.

Algorithm Al1, is similar to A but, using an (approximate) "equivalent measurement" for the measurements in the interval $[k-l+1,k]$, its second term is a linear transformation of the innovation corresponding to the equivalent measurement. Algorithm Bl1 uses, similarly to B, only the first term from Al1 and it does not need the equivalent measurement.

As shown in [3], both algorithms, while suboptimal, performed within 1% of the optimum obtained by reordering and reprocessing the measurements, which would not be practical in real systems. In view of their performance and the fact that, in the presence of biases, the statistical relationship between the "equivalent measurement" and the biases is difficult to quantify, the proposed approach is to modify Bl1 to account for the biases.

The suboptimal technique Bl1 [3] assumes the retrodicted noise to be zero. The retrodiction of the state to κ from k is[3]

$$
\hat{x}(\kappa \mid k) = F(\kappa,k)\hat{x}(k \mid k) \tag{69}
$$

i.e., a linear function of $\hat{x}(k \mid k)$, rather than an affine function. The covariance of this state retrodiction is

$$
\begin{aligned}
P_{xx}(\kappa \mid k) = {} & F(\kappa,k)[P_{xx}(k \mid k) + P_{vv}(k,\kappa \mid k) - P_{xv}(k,\kappa \mid k) \\
& - P_{xv}(k,\kappa \mid k)']F(\kappa,k)'
\end{aligned} \tag{70}
$$

where

$$
P_{vv}(k,\kappa \mid k) = Q(k,\kappa) \tag{71}
$$

$$
P_{xv}(k,\kappa \mid k) = P_{xx}(k \mid k)P_{xx}(k \mid k-l)^{-1}Q(k,\kappa) \tag{72}
$$

are the covariances of the process noise for the retrodiction interval and its crosscovariance with the current state, respectively. Equation (72) above follows by substituting in Equation (37) of [3] its preceding Equations (24) and (18) and simplifying the result.

The covariance of the retrodicted measurement, as given in Equation (39) of [3] for the situation *without biases*, is, assuming the OOSM is from sensor i, given

[3]The superscript ^{B} used in [3] to distinguish between the variables in algorithm versions A and B is dropped, since here we use only algorithm B.

by

$$S^i(\kappa) = H_x^i(\kappa)P(\kappa \mid k)H_x^i(\kappa)' + R^i(\kappa). \tag{73}$$

For the situation of the state augmented with biases, (73) is replaced by

$$\begin{aligned} S^i(\kappa) &= \mathbf{H}^i(\kappa)\mathbf{P}(\kappa \mid k)\mathbf{H}^i(\kappa)' + R^i(\kappa) \\ &= H_x^i(\kappa)P_{xx}(\kappa \mid k)H_x^i(\kappa)' + H_x^i(\kappa)P_{xb^i}(\kappa \mid k)H_b^i(\kappa)' \\ &\quad + H_b^i(\kappa)P_{b^ix}(\kappa \mid k)H_x^i(\kappa)' + H_b^i(\kappa)P_{b^ib^i}H_b^i(\kappa)' + R^i(\kappa) \end{aligned} \tag{74}$$

where, using (8), (69), one has

$$\begin{aligned} P_{xb^i}(\kappa \mid k) &= E\{[x(\kappa) - \hat{x}(\kappa \mid k)][b^i]'\} \\ &= E\{[F(\kappa,k)[x(k) - v(k,\kappa)] - F(\kappa,k)\hat{x}(k \mid k)][b^i]'\} \\ &= F(\kappa,k)P_{xb^i}(k \mid k) \end{aligned} \tag{75}$$

because the residual bias and the process noise are independent.

The crosscovariance between the state at k and the OOSM is, for the case without biases, given by Equation (40) of [3] as

$$P_{xz^i}(k,\kappa \mid k) = [P_{xx}(k \mid k) - P_{xv}(k,\kappa \mid k)]F(\kappa,k)'H_x^i(\kappa)'. \tag{76}$$

In the case with biases one has

$$\begin{aligned} P_{xz^i}(k,\kappa \mid k) &= E\{[x(k) - \hat{x}(k \mid k)][z^i(\kappa) - \hat{z}^i(\kappa \mid k)]'\} \\ &= E\{[x(k) - \hat{x}(k \mid k)][H_x^i(\kappa)x(\kappa) + w^i(\kappa) + H_b^i(\kappa)b^i - H_x^i(\kappa)F(\kappa,k)\hat{x}(k \mid k)]'\} \\ &= E\{[x(k) - \hat{x}(k \mid k)][H_x^i(\kappa)[F(\kappa,k)(x(k) - v(k,\kappa))] + w^i(\kappa) + H_b^i(\kappa)b^i - H_x^i(\kappa)F(\kappa,k)\hat{x}(k \mid k)]'\} \\ &= [P_{xx}(k \mid k) - P_{xv}(k,\kappa \mid k)]F(\kappa,k)'H_x^i(\kappa)' + P_{xb^i}(k \mid k)H_b^i(\kappa)'. \end{aligned} \tag{77}$$

Therefore, the gain for the update of the current state estimate with the OOSM $z^i(\kappa)$ in the presence of biases is (the first block of $\mathbf{W}^i(k,\kappa)^{\mathrm{OPT}} = P_{\mathbf{x}z^i}(k,\kappa \mid k)S^i(\kappa)^{-1}$)

$$W_x^i(k,\kappa) = P_{xz^i}(k,\kappa \mid k)S^i(\kappa)^{-1} \tag{78}$$

with $P_{xz^i}(k,\kappa \mid k)$ given in (77) and $S^i(\kappa)$ given in (74).

The update with the OOSM $z^i(\kappa)$ of the most recent state estimate $\hat{x}(k \mid k)$ is thus

$$\hat{x}(k \mid \kappa) = \hat{x}(k \mid k) + W_x^i(k,\kappa)\nu^i(\kappa) \tag{79}$$

where the innovation corresponding to the OOSM $z^i(\kappa)$ is

$$\nu^i(\kappa) = z^i(\kappa) - \hat{z}^i(\kappa \mid k) \tag{80}$$

and the retrodicted OOSM is

$$\hat{z}^i(\kappa \mid k) = H_x^i(\kappa)\hat{x}(\kappa \mid k) \tag{81}$$

which uses the retrodicted state $\hat{x}(\kappa \mid k)$ given in (69). Using the filter gain given in (78) and the (approximate) crosscovariance in (72), the covariance for the state estimate and the crosscovariances of the state with the biases can be obtained from (66)–(68).

As it can be seen from (72), the need to invert the state covariance and the augmentation of the state with all the sensor biases would make the algorithm prohibitive for real-time implementation. The procedure presented above avoids the need to invert the augmented covariance matrix since it does not use any state augmentation.

6. THE IMM ESTIMATOR IN THE PRESENCE OF MEASUREMENT BIASES

As discussed above, one can carry out target state estimation with biased measurements from multiple sensors without augmenting the state with all the biases. This was shown in the context of Kalman filtering, i.e., when a single target motion model is used. Next, these results are extended to the case where multiple motion models are used and the tracking filter is an IMM estimator [1]. In this case, because of the biases, each of the r modules of the IMM will be an SKF.

6.1. Update With A Current Measurement

In the IMM algorithm, the first step is mixing. Using the augmented representation, the mixed state and mixed covariance are given by

$$\begin{aligned} &\hat{\mathbf{x}}_m^0(k-1 \mid k-1) \\ &\quad = \sum_{n=1}^{r} \hat{\mathbf{x}}_n(k-1 \mid k-1)\mu_{n|m}(k-1 \mid k-1) \end{aligned} \tag{82}$$

$$\begin{aligned} &\mathbf{P}_m^0(k-1 \mid k-1) \\ &\quad = \sum_{n=1}^{r} \mu_{n|m}(k-1 \mid k-1) \\ &\qquad \cdot \{\mathbf{P}_n(k-1 \mid k-1) \\ &\qquad + [\hat{\mathbf{x}}_n(k-1 \mid k-1) - \hat{\mathbf{x}}_m^0(k-1 \mid k-1)] \\ &\qquad \cdot [\hat{\mathbf{x}}_n(k-1 \mid k-1) - \hat{\mathbf{x}}_m^0(k-1 \mid k-1)]'\} \end{aligned} \tag{83}$$

for $m = 1,\ldots,r$, where $\mu_{n|m}$ is the mixing probability [1], and the augmented state estimate $\hat{\mathbf{x}}_n(k-1 \mid k-1)$ and state covariance $\mathbf{P}_n(k-1 \mid k-1)$ matched to mode n are

$$\hat{\mathbf{x}}_n(k-1 \mid k-1) = \begin{bmatrix} \hat{x}_n(k-1 \mid k-1) \\ 0 \\ 0 \end{bmatrix} \tag{84}$$

$$\mathbf{P}_n(k-1 \mid k-1) = \begin{bmatrix} P_{x_n x_n}(k-1 \mid k-1) & P_{x_n b^i}(k-1 \mid k-1) & P_{x_n \beta^j}(k-1 \mid k-1) \\ P_{x_n b^i}(k-1 \mid k-1)' & P_{b^i b^i} & 0 \\ P_{x_n \beta^j}(k-1 \mid k-1)' & 0 & P_{\beta^j \beta^j} \end{bmatrix} \tag{85}$$

with the bias terms in (84) being zero in view of (23). Using (82)–(85), the blocks of (82) and (83) are obtained as

$$\hat{x}_m^0(k-1 \mid k-1) = \sum_{n=1}^{r} \hat{x}_n(k-1 \mid k-1)\mu_{n|m}(k-1 \mid k-1) \tag{86}$$

$$\begin{aligned} P_{x_m x_m}^0(k-1 \mid k-1) &= \sum_{n=1}^{r} \mu_{n|m}(k-1 \mid k-1) \\ &\quad \cdot \{P_{x_n x_n}(k-1 \mid k-1) \\ &\qquad + [\hat{x}_n(k-1 \mid k-1) - \hat{x}_m^0(k-1 \mid k-1)] \\ &\qquad \cdot [\hat{x}_n(k-1 \mid k-1) - \hat{x}_m^0(k-1 \mid k-1)]'\} \end{aligned} \tag{87}$$

$$P_{x_m b^i}^0(k-1 \mid k-1) = \sum_{n=1}^{r} \mu_{n|m}(k-1 \mid k-1) P_{x_n b^i}(k-1 \mid k-1) \tag{88}$$

$$P_{x_m \beta^j}^0(k-1 \mid k-1) = \sum_{n=1}^{r} \mu_{n|m}(k-1 \mid k-1) P_{x_n \beta^j}(k-1 \mid k-1) \qquad \forall j \neq i(k). \tag{89}$$

The likelihood of mode m at time k is, assuming the mode-conditioned innovations to be Gaussian distributed (the common assumption [1]) is

$$\Lambda_m(k) = \mathcal{N}[\nu_m^i(k); 0, S_m^i(k)], \qquad m = 1,\ldots,r \tag{90}$$

where, with $\hat{x}_m(k \mid k-1)$ being the mode-m-conditioned predicted state, the innovation corresponding to mode m is, similarly to (34),

$$\nu_m^i(k) = z^i(k) - H_x^i(k)\hat{x}_m(k \mid k-1) \tag{91}$$

and the innovation covariance is, similarly to (25),

$$\begin{aligned} S_m^i(k) &= H_x^i(k) P_{x_m x_m}(k \mid k-1) H_x^i(k)' \\ &\quad + H_x^i(k) P_{x_m b^i}(k \mid k-1) H_b^i(k)' \\ &\quad + H_b^i(k) P_{b^i x_m}(k \mid k-1) H_x^i(k)' \\ &\quad + H_b^i(k) P_{b^i b^i} H_b^i(k)' + R^i(k) \end{aligned} \tag{92}$$

where $P_{x_m x_m}(k \mid k-1)$ and $P_{x_m b^i}(k \mid k-1)$ are the predicted state covariance and state-bias crosscovariance matched to mode m. In the above it is assumed that the measurement equations are the same for all the modes. The values of $\hat{x}_m(k \mid k-1)$, $P_{x_m x_m}(k \mid k-1)$, and $P_{x_m b^i}(k \mid k-1)$ are obtained from (35)–(39) using the mixed state estimate and mixed covariances given in (86)–(89).

Based on the mode likelihoods, the model probabilities at current time k, $\{\mu_m(k \mid k)\}_{m=1,\ldots,r}$, can be obtained [1], which are then used to calculate the combined state estimate and state covariance, namely,

$$\hat{\mathbf{x}}(k \mid k) = \sum_{m=1}^{r} \hat{\mathbf{x}}_m(k \mid k)\mu_m(k \mid k) \tag{93}$$

$$\begin{aligned} \mathbf{P}(k \mid k) &= \sum_{m=1}^{r} \mu_m(k \mid k) \\ &\quad \cdot \{\mathbf{P}_m(k \mid k) + [\hat{\mathbf{x}}_m(k \mid k) - \hat{\mathbf{x}}(k \mid k)] \\ &\qquad \cdot [\hat{\mathbf{x}}_m(k \mid k) - \hat{\mathbf{x}}(k \mid k)]'\}. \end{aligned} \tag{94}$$

Similar to the mixing step, the blocks of (93) and (94) are obtained as

$$\hat{x}(k \mid k) = \sum_{m=1}^{r} \hat{x}_m(k \mid k)\mu_m(k \mid k) \tag{95}$$

$$\begin{aligned} P_{xx}(k \mid k) &= \sum_{m=1}^{r} \mu_m(k \mid k) \\ &\quad \cdot \{P_{x_m x_m}(k \mid k) + [\hat{x}_m(k \mid k) - \hat{x}(k \mid k)] \\ &\qquad \cdot [\hat{x}_m(k \mid k) - \hat{x}(k \mid k)]'\} \end{aligned} \tag{96}$$

$$P_{xb^i}(k \mid k) = \sum_{m=1}^{r} \mu_m(k \mid k) P_{x_m b^i}(k \mid k) \tag{97}$$

$$P_{x\beta^j}(k \mid k) = \sum_{m=1}^{r} \mu_m(k \mid k) P_{x_m\beta^j}(k \mid k), \qquad \forall j \neq i(k) \tag{98}$$

where the values of $\hat{x}_m(k \mid k)$, $P_{x_m x_m}(k \mid k)$, $P_{x_m b^i}(k \mid k)$, and $P_{x_m\beta^j}(k \mid k)$ are obtained from (27)–(33) for each mode.

6.2. Update With An OOSM

Using the same notations—subscripting with m the mode-m-conditioned state estimates, covariances, innovations—the equations from Section 5 provide the procedure for using an OOSM in each module. The likelihood of each mode based on an OOSM will be, analogously to (90),

$$\Lambda_m(\kappa) = \mathcal{N}[\nu_m^i(\kappa); 0, S_m^i(\kappa)], \qquad m = 1,\ldots,r \tag{99}$$

where, with $\hat{x}_m(\kappa \mid k)$ being the mode-m-conditioned retrodicted state, the innovation corresponding to mode m is, similarly to (80),

$$\nu_m^i(\kappa) = z^i(\kappa) - H_x^i(\kappa)\hat{x}_m(\kappa \mid k) \tag{100}$$

where $\hat{x}_m(\kappa \mid k)$ is given by

$$\hat{x}_m(\kappa \mid k) = F_m(\kappa, k)\hat{x}_m(k \mid k). \tag{101}$$

The innovation covariance is, similarly to (74),

$$\begin{aligned} S_m^i(\kappa) &= H_x^i(\kappa) P_{x_m x_m}(\kappa \mid k) H_x^i(\kappa)' + H_x^i(\kappa) P_{x_m b^i}(\kappa \mid k) H_b^i(\kappa)' \\ &\quad + H_b^i(\kappa) P_{b^i x_m}(\kappa \mid k) H_x^i(\kappa)' + H_b^i(\kappa) P_{b^i b^i} H_b^i(\kappa)' + R^i(\kappa). \end{aligned} \tag{102}$$

Using the likelihood function (99), the current mode probabilities updated with the OOSM are as in [4], namely,

$$\mu_m(k \mid \kappa) = \frac{1}{c}\left[\sum_{n=1}^{r} \Lambda_n(\kappa)\Pi_{mn}(\kappa, k)\right]\mu_m(k \mid k). \tag{103}$$

where the normalization constant is

$$c = \sum_{m=1}^{r}\sum_{n=1}^{r} \Lambda_n(\kappa)\Pi_{mn}(\kappa, k)\mu_m(k \mid k). \tag{104}$$

The mode transition probability $\Pi_{mn}(k_2, k_1)$ from time k_1 to k_2 is defined as

$$\Pi_{mn}(k_2, k_1) = P\{M(k_2) = n \mid M(k_1) = m\} \tag{105}$$

which is an element (row m, column n) in the transition matrix $\Pi(k_2, k_1)$. For $r = 2$, the transition matrix $\Pi(k_2, k_1)$ according to a continuous-time Markov chain is given by [4]

$$\Pi(k_2, k_1) = \frac{1}{\lambda_1 + \lambda_2}\begin{bmatrix} \lambda_2 + \lambda_1 e^{-(\lambda_1+\lambda_2)T} & \lambda_1 - \lambda_1 e^{-(\lambda_1+\lambda_2)T} \\ \lambda_2 - \lambda_2 e^{-(\lambda_1+\lambda_2)T} & \lambda_1 + \lambda_2 e^{-(\lambda_1+\lambda_2)T} \end{bmatrix} \tag{106}$$

where $T = |t_{k_2} - t_{k_1}|$, and $1/\lambda_m$ is the expected sojourn time for mode m.

Similar to (95)–(98), the combined state estimate and covariances with the OOSM are given by

$$\hat{x}(k \mid \kappa) = \sum_{m=1}^{r} \hat{x}_m(k \mid \kappa)\mu_m(k \mid \kappa) \tag{107}$$

$$\begin{aligned} P_{xx}(k \mid \kappa) &= \sum_{m=1}^{r} \mu_m(k \mid \kappa) \\ &\quad \cdot \{P_{x_m x_m}(k \mid \kappa) + [\hat{x}_m(k \mid \kappa) - \hat{x}(k \mid \kappa)] \\ &\quad \cdot [\hat{x}_m(k \mid \kappa) - \hat{x}(k \mid \kappa)]'\} \end{aligned} \tag{108}$$

$$P_{xb^i}(k \mid \kappa) = \sum_{m=1}^{r} \mu_m(k \mid \kappa) P_{x_m b^i}(k \mid \kappa) \tag{109}$$

$$P_{x\beta^j}(k \mid \kappa) = \sum_{m=1}^{r} \mu_m(k \mid \kappa) P_{x_m\beta^j}(k \mid \kappa), \qquad \forall j \neq i(k) \tag{110}$$

where the values of $\hat{x}_m(k \mid \kappa)$, $P_{x_m x_m}(k \mid \kappa)$, $P_{x_m b^i}(k \mid \kappa)$, and $P_{x_m\beta^j}(k \mid \kappa)$ are obtained from (79) and (66)–(68) for each mode. Note that the mixing step is not carried out with the OOSM [4].

7. THE HEURISTIC "COVARIANCE INFLATION" APPROACH FOR BIASES

This approach increases the measurement noise variance by the variance of the biases (assumed to have mean zero). Note that this amounts to treating the biases as if they were an additional zero-mean white noise, which is clearly not correct. Only the SKF correctly treats the biases as fixed random variables. The reason this heuristic approach is considered here is that it has been used due to its simplicity, but as it will be shown, it yields inconsistent estimates.

Consider a measurement model given by

$$z(k) = h(x(k), b) + w(k) \tag{111}$$

which may be a nonlinear function of the target state and residual biases. The superscript i has been dropped for simplicity. Using the first-order Taylor expansion at $x(k) = \hat{x}(k \mid k-1)$ and $b = 0$, the measurement $z(k)$ can be approximated as

$$\begin{aligned} z(k) &\approx h(\hat{x}(k \mid k-1), 0) + H_x(k)(x(k) - \hat{x}(k \mid k-1)) \\ &\quad + H_b(k)(b - 0) + w(k) \\ &= h(\hat{x}(k \mid k-1), 0) + H_x(k)\tilde{x}(k \mid k-1) \\ &\quad + H_b(k)b + w(k) \end{aligned} \tag{112}$$

where

$$H_x(k) = \left.\frac{\partial h(x,b)}{\partial x}\right|_{x(k)=\hat{x}(k|k-1), b=0} \tag{113}$$

$$H_b(k) = \left.\frac{\partial h(x,b)}{\partial b}\right|_{x(k)=\hat{x}(k|k-1), b=0} \tag{114}$$

TABLE I
Sensor Indices and Corresponding Time Stamps

Sensor Index	1	1	2	1	2	1	2	1	2	1	2	1	2	1
Time Stamp (s)	0	5	2.5	10	7.5	15	12.5	20	17.5	25	22.5	30	27.5	35

and $\hat{x}(k \mid k-1)$ is the state prediction at time k using the set of measurements z^{k-1}. The MMSE estimate of $z(k)$ is given by

$$\hat{z}(k \mid k-1) = E\{z(k) \mid z^{k-1}\} \approx h(\hat{x}(k \mid k-1), 0) \tag{115}$$

which has used the Taylor approximation given in (112) (this is actually the estimate using EKF). The corresponding error covariance, which has the same expression as in (25), is

$$\begin{aligned} S(k) &= \mathrm{cov}\{z(k) - \hat{z}(k \mid k-1)\} \\ &\approx \mathrm{cov}\{H_x(k)\tilde{x}(k \mid k-1) + H_b(k)b + w(k)\} \\ &= H_x(k)\mathrm{cov}\{\tilde{x}(k \mid k-1)\}H_x'(k) \\ &\quad + H_b(k)\mathrm{cov}\{b\}H_b'(k) + \mathrm{cov}\{w(k)\} \\ &\quad + H_x(k)\mathrm{cov}\{\tilde{x}(k \mid k-1), b\}H_b'(k) \\ &\quad + H_b(k)\mathrm{cov}\{b, \tilde{x}(k \mid k-1)\}H_x'(k) \end{aligned} \tag{116}$$

due to the fact that the measurement noise $w(k)$ is independent of the prediction error $\tilde{x}(k \mid k-1)$ and residual biases b. If the crosscovariances between the estimation error and residual biases are set to be zero, $S(k)$ can be written as

$$\begin{aligned} S(k) &= H_x(k)\mathrm{cov}\{\tilde{x}(k \mid k-1)\}H_x'(k) + H_b(k)\mathrm{cov}\{b\}H_b'(k) \\ &\quad + \mathrm{cov}\{w(k)\} \end{aligned} \tag{117}$$

which amounts to a covariance inflation with the inflation term $H_b(k)\mathrm{cov}\{b\}H_b'(k)$, similar to that in [18]. The effect of ignoring the crosscovariances will be evaluated in the simulation results.

8. SIMULATION RESULTS

8.1. Example 1: One-Dimensional Motion With Position Measurement Only

The target starts at origin and moves with a constant velocity of 10 m/s along the x-axis. The power spectrum density (PSD) of the process noise is $q = 0.5\ \mathrm{m}^2/\mathrm{s}^3$. Two sensors are used, which are located at $(-50, 0)$ km and $(50, 0)$ km. The (unaugmented) target state is denoted by $\mathbf{x}$ as

$$\mathbf{x} = [x \quad \dot{x}] \tag{118}$$

and the measurement model is

$$z^i = (1 + \alpha^i)[1 \quad 0](\mathbf{x} - \mathbf{x}_p^i) + \Delta^i + w^i, \qquad i = 1, 2 \tag{119}$$

TABLE II
Bias Standard Deviations for Position Measurement (Example 1)

Bias Level	Offset Bias Δ	Scale Bias α
Small	10 m ($= \sigma_w$)	10^{-4}
Large	20 m ($= 2\sigma_w$)	2×10^{-4}

where Δ denotes an offset bias and α denotes a scale (multiplicative) bias. The superscript i is the sensor index and $\mathbf{x}_p^i$ is the state of the ith sensor. The measurement noise s.d. is $\sigma_w = 10$ m for both sensors. The sampling interval for each sensor is 5 s, but they are not synchronized. The times at which the measurements are taken (their "time stamps") and the order of the measurements arriving at the fusion center are shown in Table I, where sensor 2 has all its measurements delayed with 1 step lag.

Two levels of biases are considered with the bias s.d. given in Table II (the same for both sensors).

For each bias level, two options are considered:

1. Reorder the measurements (in-sequence data).
2. Process OOSM.

The option of ignoring OOSMs has been shown in [3] to lead to significant performance loss.

For each scenario, we compare three filters: Kalman filter[4] without covariance inflation (KFwoINF), Kalman filter with covariance inflation (KFwINF), and SKF (Schmidt-KF). Each of these is modified appropriately when processing OOSM. The modified SKF to process OOSM is the SKF/OOSM from Section 5.

The discretized CWNA model (DCWNA) [1] is used as the target's dynamic model. Since the smallest time interval between the time stamps shown in Table I is 2.5 s, we use $T = 2.5$ s as the filter's sampling interval. The results below are based on 1000 Monte Carlo simulations. The two-sided 99% probability region of the NEES (normalized estimation error squared, [1] Section 5) based on the χ^2_{2000} distribution [1] is $[1.84, 2.16]$, marked by two dashed lines in the figures.

All filters were initialized with "one point" initialization (see [1], Section 5) according to which the first position measurement is used as the initial estimate and the initial velocity estimate is set to zero; the standard deviation of the latter is set at half the maximum speed in each coordinate.

The RMSE at the times when OOSMs are processed (at the time stamps of sensor 1 in our case) in Figs. 3

[4]Here, "Kalman filter" may also refer to extended Kalman Filter for nonlinear cases. We use the same acronym "KF" for simplicity.

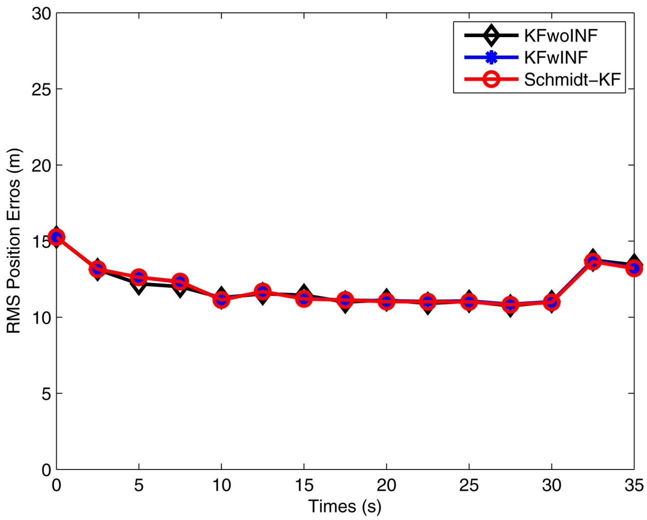


Fig. 1. Position RMSE for target with position measurement only (small bias, reordering measurements).

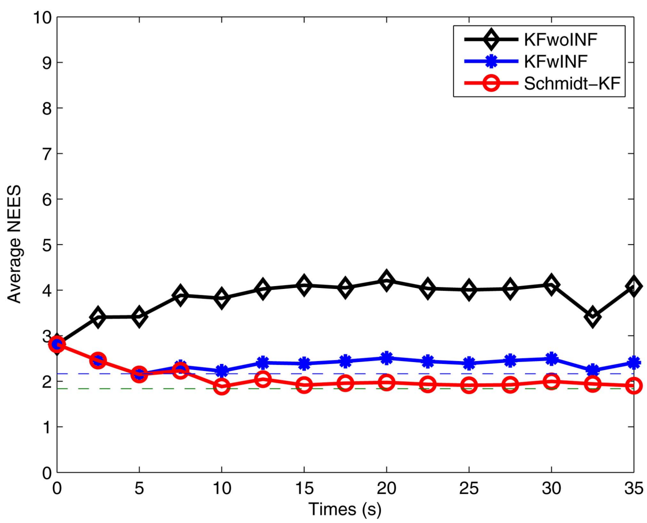


Fig. 2. NEES for target with position measurement only (small bias, reordering measurements).

and 7, is nearly the same as the RMSE from the in-sequence data in Figs. 1 and 5 at the corresponding times. From the NEES in Figs. 2, 4, 6 and 8, we can see that SKF is consistent (i.e., its NEES falls in its probability region [1]) for both in-sequence data and processing OOSMs. KFwoINF is the most inconsistent (overly optimistic) and its inconsistency increases with the bias level. KFwINF improves the filter's consistency but it is still not consistent due to ignoring the correlation between the bias and estimation error (because it assumes the bias to be white noise, as indicated in Section 7). The inconsistency of KFwINF also increases with the bias level. The results also show that KFwINF and SKF do not improve the estimation accuracy in this example with position only measurements. However, in the next example we will see that the SKF does improve the estimation accuracy for the case of GMTI measurements, which include additional range rate measurements.

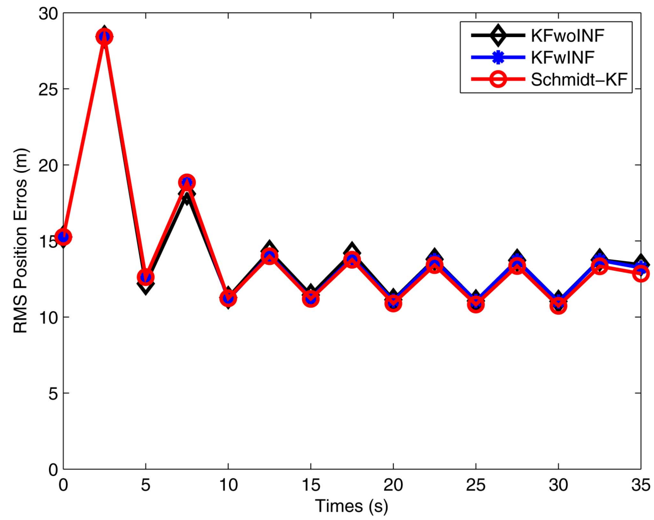


Fig. 3. Position RMSE for target with position measurement only (small bias, OOSM processing).

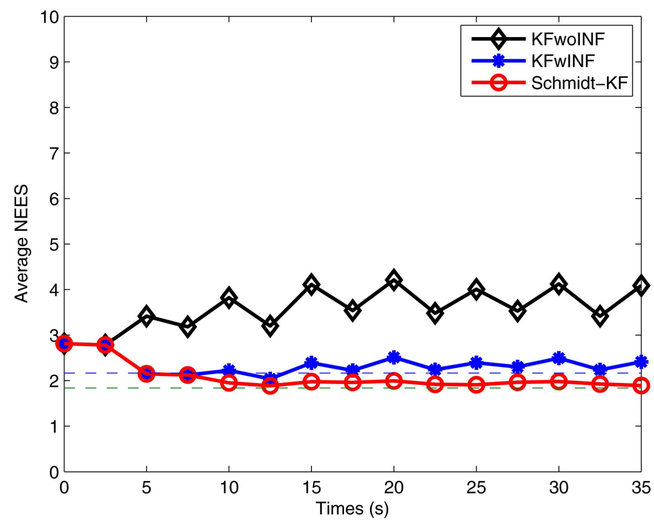


Fig. 4. NEES for target with position measurement only (small bias, OOSM processing).

Tables III and IV show a comparison of the three algorithms for various levels of process noise in the case of small bias for in-sequence data and processing OOSMs, respectively. SKF is consistent in almost all cases.[5] For all process noise PSD levels, KFwoINF is inconsistent (optimistic). For KFwINF, when the PSD is around 10 $\mathrm{m}^2/\mathrm{s}^3$ (with the maneuvering index around 1), KFwINF is consistent and for the other cases, KFwINF is inconsistent.

For large bias, the results are shown in Tables V and VI for in-sequence data and processing OOSMs, respectively. The consistency of SKF is the best, though it seems a little "pessimistic" when the process noise PSD is below 0.01 $\mathrm{m}^2/\mathrm{s}^3$. For this case of large bias, KFwINF is always inconsistent. Also, from the RMS

[5] Since the multiplicative bias requires linearization, minor inconsistencies (NEES slightly outside the probability region) can occur.

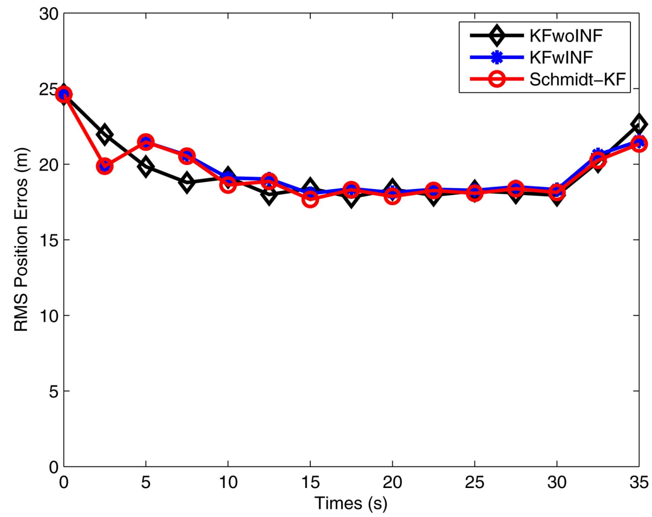


Fig. 5. Position RMSE for target with position measurement only (large bias, reordering measurements).

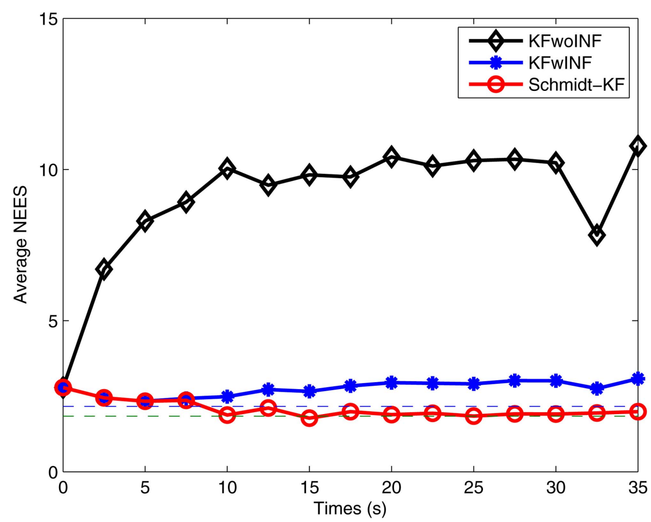


Fig. 6. NEES for target with position measurement only (large bias, reordering measurements).

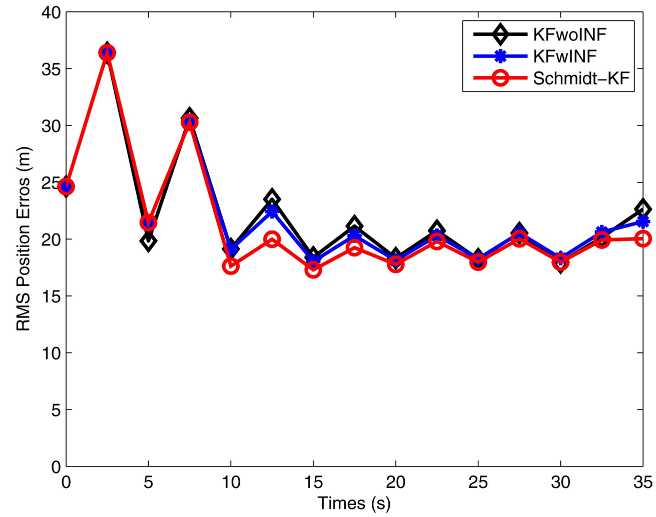


Fig. 7. Position RMSE for target with position measurement only (large bias, OOSM processing).

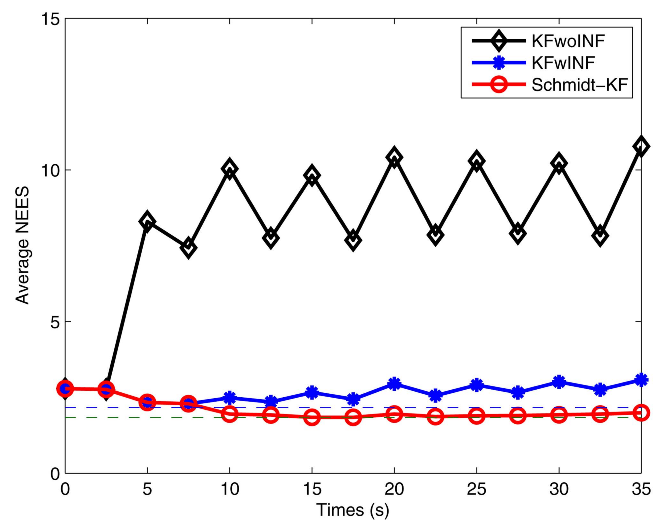


Fig. 8. NEES for target with position measurement only (large bias, OOSM processing).

in Tables III–VI we can notice a small improvement in estimation accuracy using SKF at the final estimate (as in Figs. 1, 3, 5 and 7). However, as shown in Figs. 1, 3, 5 and 7, this improvement is not significant and may not be achieved for every point.

In a realistic scenario, the target is usually tracked in 3D space and a more complicated bias model should be used [15]. However, the results with the use of SKF are similar to the above 1D case and this simple example provides a good illustration of the effect of biases.

8.2. Example 2: Target With Low Process Noise And GMTI Measurements

This example considers a target that moves in a 2-dimensional space with a nearly constant velocity. The target state consists of position and velocity along each coordinate (x and y). The initial target state is $[100\text{ m}, 9\text{ m/s}, 200\text{ m}, 5\text{ m/s}]$. The PSD of the process noise is $q = 0.5\text{ m}^2/\text{s}^3$ for both x and y coordinate. The motion model considered is DCWNA. Two GMTI radars are located with nearly perpendicular LOS to the target. One is at $(-48, 13)$ km with a slant range around 50 km to the target and the other is at $(-26, -96)$ km with a slant range around 100 km to the target. The measurements are range (r), azimuth (θ) and range rate ($\dot{r}$) with s.d. $\sigma_r = 10$ m, $\sigma_\theta = 1$ mrad and $\sigma_{\dot{r}} = 1$ m/s, respectively, for both sensors. The measurement model is

$$z^i = (I_3 + \Lambda^i)h(\mathbf{x} - \mathbf{x}_p^i) + \Delta^i + w^i, \qquad i = 1,2 \tag{120}$$

where

$$\mathbf{x} = [x \quad \dot{x} \quad y \quad \dot{y}]' \tag{121}$$

denotes here the (unaugmented) target state and and $\mathbf{x}_p^i$ denotes the ith sensor state, the function $h : \mathcal{R}^4 \to \mathcal{R}^3$ is

TABLE III
Comparison of RMS Errors at Final Estimate for Different Maneuvering Index Based on 1000 Monte Carlo Runs
(Small Bias, Reordering Measurements)

Algorithm	Maneuvering Index λ	Process Noise PSD (m^2/s^3)	Position RMS (m)	Velocity RMS (m/s)	NEES
KFwoINF	0.0125	0.001	10.4110	0.3176	9.6063
KFwINF	0.0125	0.001	10.2422	0.2914	4.8272
Schmidt-Kalman	0.0125	0.001	10.1193	0.2832	1.9872
KFwoINF	0.0395	0.01	10.5154	0.4651	6.2039
KFwINF	0.0395	0.01	10.3116	0.4382	3.4659
Schmidt-Kalman	0.0395	0.01	10.2021	0.4299	1.8121
KFwoINF	0.125	0.1	11.7715	1.0856	4.7297
KFwINF	0.125	0.1	11.4237	1.0669	2.8152
Schmidt-Kalman	0.125	0.1	11.3450	1.0640	1.9334
KFwoINF	0.3953	1	14.0554	2.2643	4.0057
KFwINF	0.3953	1	13.9014	2.2850	2.3912
Schmidt-Kalman	0.3953	1	13.8626	2.2634	1.9900
KFwoINF	1.25	10	14.0609	4.8089	3.3235
KFwINF	1.25	10	14.1587	4.9487	2.0359
Schmidt-Kalman	1.25	10	13.9012	4.8435	1.8791

TABLE IV
Comparison of RMS Errors at Final Estimate for Different Maneuvering Index Based on 1000 Monte Carlo Runs
(Small Bias, OOSM Processing)

Algorithm	Maneuvering Index λ	Process Noise PSD (m^2/s^3)	Position RMS (m)	Velocity RMS (m/s)	NEES
KFwoINF	0.0125	0.001	10.4110	0.3176	9.6063
KFwINF	0.0125	0.001	10.2422	0.2914	4.8271
Schmidt-Kalman	0.0125	0.001	10.0202	0.2829	2.0357
KFwoINF	0.0395	0.01	10.5155	0.4651	6.2039
KFwINF	0.0395	0.01	10.3116	0.4382	3.4658
Schmidt-Kalman	0.0395	0.01	10.0704	0.4254	1.8400
KFwoINF	0.125	0.1	11.7723	1.0856	4.7301
KFwINF	0.125	0.1	11.4239	1.0669	2.8153
Schmidt-Kalman	0.125	0.1	11.0296	1.0564	1.9419
KFwoINF	0.3953	1	14.0557	2.2639	4.0057
KFwINF	0.3953	1	13.9006	2.2848	2.3909
Schmidt-Kalman	0.3953	1	13.5495	2.2835	1.9692
KFwoINF	1.25	10	14.0624	4.8086	3.3233
KFwINF	1.25	10	14.1590	4.9481	2.0351
Schmidt-Kalman	1.25	10	13.8653	4.8579	1.8629

given by

$$h(\mathbf{x}) = \begin{bmatrix} r \\ \theta \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \sqrt{x^2+y^2} \\ \tan^{-1}\dfrac{y}{x} \\ \dot{x}\cos\theta + \dot{y}\sin\theta \end{bmatrix}. \qquad (122)$$

I_3 denotes the 3×3 identity matrix, Λ represents the scale bias, having the form

$$\Lambda = \begin{bmatrix} \alpha_r & 0 & 0 \\ 0 & \alpha_\theta & 0 \\ 0 & 0 & \alpha_{\dot{r}} \end{bmatrix} \qquad (123)$$

with the bias terms on its main diagonal and Δ denotes the offset bias, that is,

$$\Delta = [\Delta_r \quad \Delta_\theta \quad \Delta_{\dot{r}}]'. \qquad (124)$$

The extended Kalman filter (EKF) is used with the Jacobian terms given in Appendix A. The order of the measurements arriving at the fusion center is shown in Table I. Two bias levels are considered, with the bias s.d. given in Table VII. The results are based on 500 Monte Carlo simulations. The two-sided 99% probability region of the NEES is [3.68, 4.33] based on the χ^2_{2000} distribution.

TABLE V

Comparison of RMS Errors at Final Estimate for Different Maneuvering Index Based on 1000 Monte Carlo Runs (Large Bias, Reordering Measurements)

Algorithm	Maneuvering Index λ	Process Noise PSD (m^2/s^3)	Position RMS (m)	Velocity RMS (m/s)	NEES
KFwoINF	0.0125	0.001	18.0346	0.4642	29.3834
KFwINF	0.0125	0.001	17.3345	0.3283	5.9046
Schmidt-Kalman	0.0125	0.001	16.8580	0.2902	1.7161
KFwoINF	0.0395	0.01	18.9462	0.6461	18.9844
KFwINF	0.0395	0.01	17.7781	0.4978	4.8489
Schmidt-Kalman	0.0395	0.01	17.2558	0.4684	1.7761
KFwoINF	0.125	0.1	20.6240	1.3104	13.7247
KFwINF	0.125	0.1	19.4403	1.1497	3.7531
Schmidt-Kalman	0.125	0.1	19.1942	1.1357	1.9113
KFwoINF	0.3953	1	23.4515	2.7203	9.7744
KFwINF	0.3953	1	22.6597	2.6533	2.7582
Schmidt-Kalman	0.3953	1	22.5325	2.6332	1.9727
KFwoINF	1.25	10	25.3687	4.9322	8.5757
KFwINF	1.25	10	25.4763	5.4990	2.3465
Schmidt-Kalman	1.25	10	24.7708	5.1617	1.9618

TABLE VI

Comparison of RMS Errors at Final Estimate for Different Maneuvering Index Based on 1000 Monte Carlo Runs (Large Bias, OOSM Processing)

Algorithm	Maneuvering Index λ	Process Noise PSD (m^2/s^3)	Position RMS (m)	Velocity RMS (m/s)NEES	
KFwoINF	0.0125	0.001	18.0346	0.4642	29.3835
KFwINF	0.0125	0.001	17.3345	0.3283	5.9043
Schmidt-Kalman	0.0125	0.001	16.5600	0.2960	1.7797
KFwoINF	0.0395	0.01	18.9465	0.6461	18.9847
KFwINF	0.0395	0.01	17.7781	0.4978	4.8488
Schmidt-Kalman	0.0395	0.01	16.7486	0.4635	1.8212
KFwoINF	0.125	0.1	20.6255	1.3103	13.7260
KFwINF	0.125	0.1	19.4405	1.1497	3.7531
Schmidt-Kalman	0.125	0.1	18.5341	1.0975	1.9393
KFwoINF	0.3953	1	23.4517	2.7192	9.7738
KFwINF	0.3953	1	22.6590	2.6530	2.7580
Schmidt-Kalman	0.3953	1	21.1264	2.6219	1.9659
KFwoINF	1.25	10	25.3720	4.9349	8.5752
KFwINF	1.25	10	25.4715	5.4991	2.3454
Schmidt-Kalman	1.25	10	24.3123	5.3158	1.9246

TABLE VII

Bias Standard Deviations for GMTI Measurements

Bias Level	Offset Bias Δ_r	Offset Bias Δ_θ	Offset Bias $\Delta_{\dot{r}}$	Scale Bias α_r	Scale Bias α_θ	Scale Bias $\alpha_{\dot{r}}$
Small	10 m ($1 \times \sigma_r$)	1 mrad ($1 \times \sigma_\theta$)	1 m/s ($1 \times \sigma_{\dot{r}}$)	1×10^{-4}	1×10^{-4}	1×10^{-4}
Large	20 m ($2 \times \sigma_r$)	2 mrad ($2 \times \sigma_\theta$)	2 m/s ($2 \times \sigma_{\dot{r}}$)	2×10^{-4}	2×10^{-4}	2×10^{-4}

From the RMSE in Figs. 9 and 11 we can see that SKF improves estimation accuracy compared to KFwoINF and KFwINF for in-sequence data as well as in case of processing OOSM. KFwINF is even worse than KFwoINF in this case. With the bias level increasing, the improvement in RMSE using SKF (shown in Figs. 13 and 15) becomes more significant. For consistency, from the NEES shown in Figs. 10 and 12 we can see that SKF takes some time to become consistent since the initial crosscovariance between the estimation

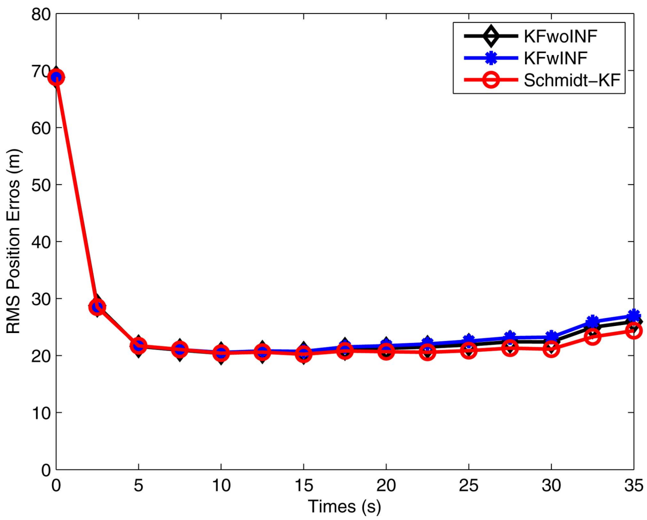


Fig. 9. Position RMSE for target with GMTI measurement (small bias, reordering measurements).

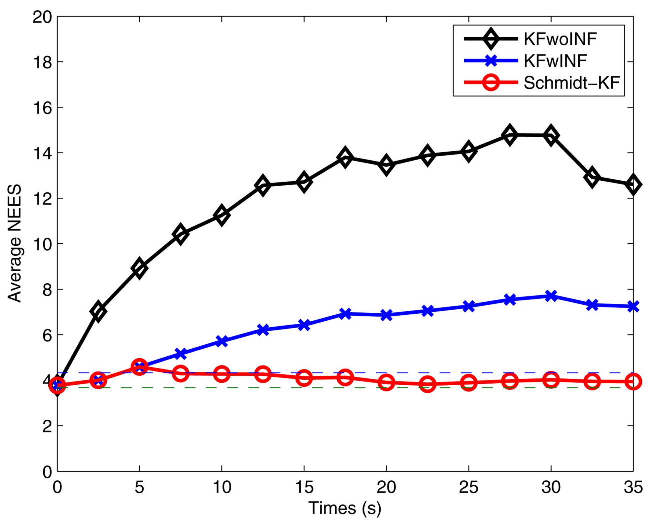


Fig. 10. NEES for target with GMTI measurement (small bias, reordering measurements).

error and the bias is set to be zero[6] and the SKF needs several updates to obtain the correct crosscovariance.

When the bias level increases, SKF needs more updates for the covariance to become consistent (as shown in Figs. 14 and 16). KFwoINF is the most inconsistent (with NEES around 40 in Fig. 14 for the case of large bias). KFwINF improves the consistency but is still not consistent (with NEES around 10 in Fig. 14 for the case of large bias).

8.3. Example 3: Maneuvering Target With GMTI Radar Measurements

This example considers a target with realistic maneuvers. The initial state of the target is [100 m,

[6]The initial crosscovariance between the estimation error and the bias is not available exactly since one needs the true state x to evaluate this crosscovariance due to the scale bias. Using the initial state estimate in the crosscovariance yields the same minor initial inconsistency as when the initial crosscovariance is set to be zero.

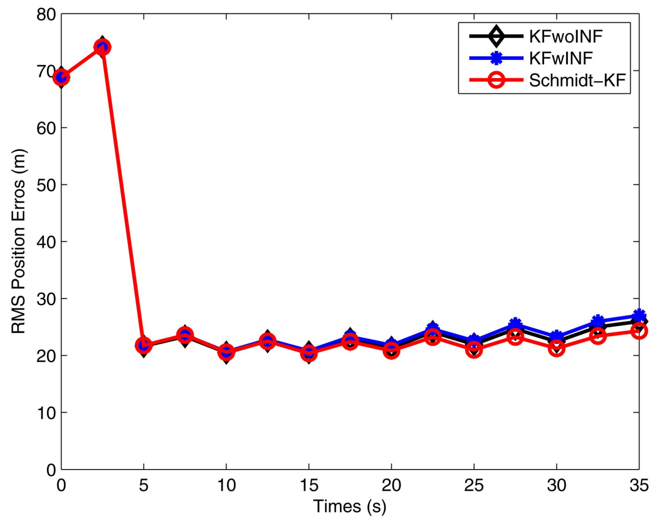


Fig. 11. Position RMSE for target with GMTI measurement (small bias, OOSM processing).

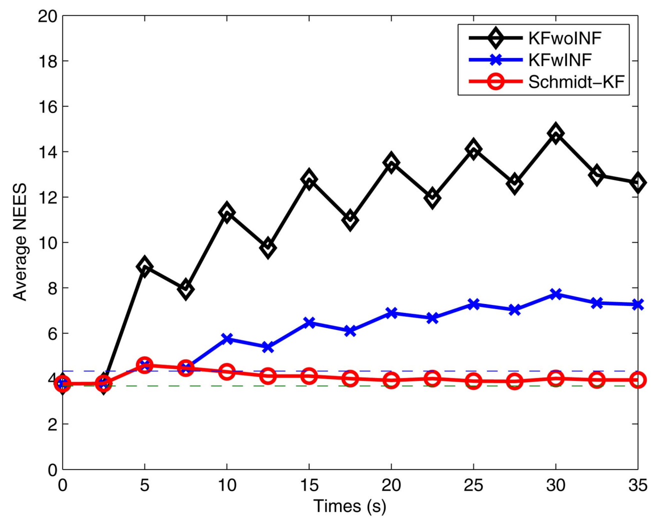


Fig. 12. NEES for target with GMTI measurement (small bias, OOSM processing).

$10/\sqrt{2}$ m/s, 200 m, $10/\sqrt{2}$ m/s]. The target moves with a constant velocity, 10 m/s during $t \in [0, 15\ \mathrm{s}]$. Then, it makes a left turn with a constant speed $V = 10$ m/s and a constant turn rate $w = 5^\circ/\mathrm{s} \approx 0.09$ rad/s during $t \in [15\ \mathrm{s}, 35\ \mathrm{s}]$. In addition to the maneuver it subjects to process noise with PSD of $q_1 = 0.01\ \mathrm{m}^2/\mathrm{s}^3$ for the entire period. The maneuver corresponds, over the sampling interval $T = 2.5$ s, to a velocity change of (approximately) $\Delta V = wVT = 2.25$ m/s. Equating this to the RMS velocity change due to the process noise over interval T, which is given by $\sqrt{q_2 T}$ [1], yields for this case $q_2 = (wVT)^2/T \approx 2\ \mathrm{m}^2/\mathrm{s}^3$.

Two GMTI radars are used in this scenario as Example 2. An IMM estimator is used to track the maneuvering target with two nearly constant velocity (NCV) [1] models. One has a low process noise PSD $q_1 = 0.01\ \mathrm{m}^2/\mathrm{s}^3$ and the other has a high process noise PSD $q_2 = 2\ \mathrm{m}^2/\mathrm{s}^3$. The mode transition matrix is (106) with

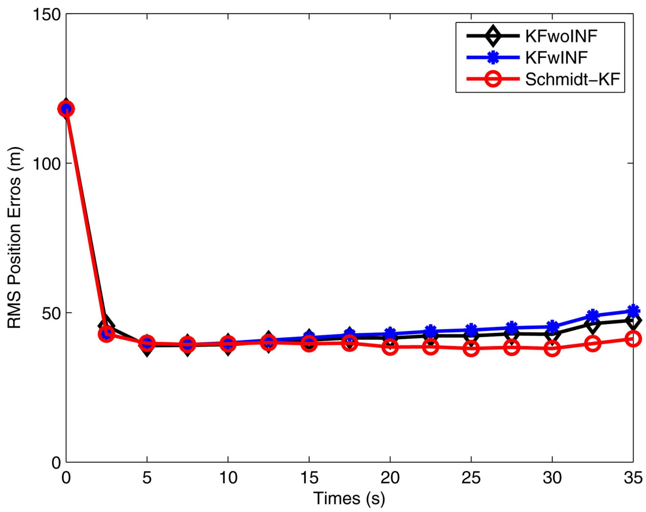


Fig. 13. Position RMSE for target with GMTI measurement (large bias, reordering measurements).

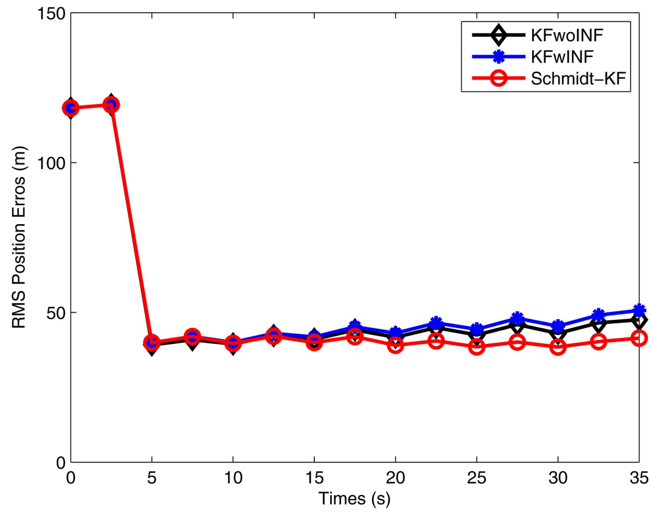


Fig. 15. Position RMSE for target with GMTI measurement (large bias, OOSM processing).

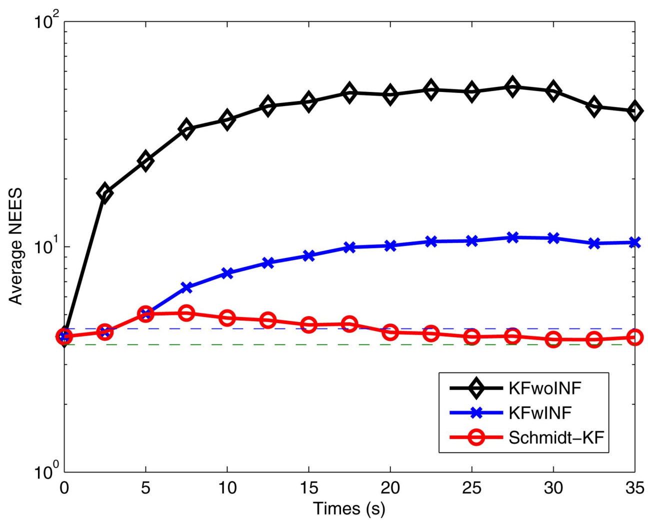


Fig. 14. NEES for target with GMTI measurement (large bias, reordering measurements).

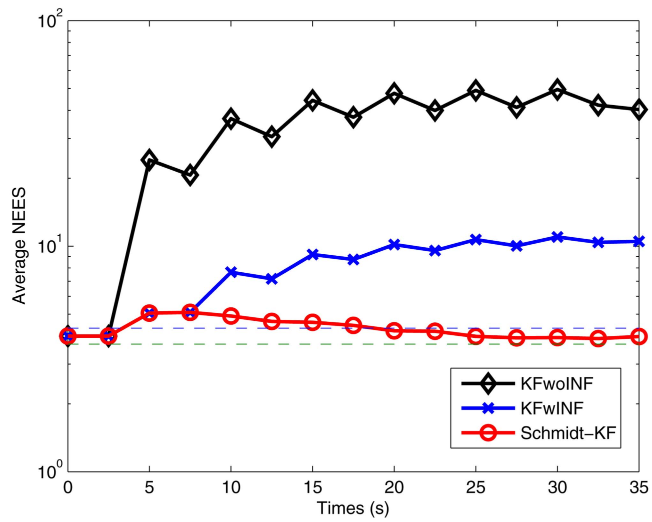


Fig. 16. NEES for target with GMTI measurement (large bias, OOSM processing).

the sojourn times: $\lambda_1^{-1} = 15$ s, $\lambda_2^{-1} = 20$ s. The motion model used is DCWNA. The measurement sequence received at the fusion center is as shown in Table I. The bias s.d. is as given in Table VII. The simulation results below are from 500 Monte Carlo runs.

From the NEES in Figs. 18, 20, 22, and 24 we can see that, due to the use of IMM filter, SKF is not consistent anymore,[7] especially during the mode transition period. However, compared to KFwoINF and KFwINF, the consistency is still improved significantly. At the times when the OOSMs are processed (at the time stamps of sensor 1 in our case), the RMSE for OOSM processing (as shown in Figs. 19 and 23) and the RMSE for in-sequence data (in Figs. 17 and 21) are almost the same. As in Example 2, SKF improves estimation accuracy compared to KFwoINF and KFwINF, even though not significantly in the case of small bias.

[7]No IMM estimation can be perfectly consistent because the inconsistency of a model drives it “soft switching”.

9. SUMMARY AND CONCLUSIONS

The single sensor algorithm Bl1, which updates the current state of a target with an OOSM from a single sensor without bias has been extended to the multisensor situation where each sensor exhibits a residual bias. This has been accomplished using the proposed algorithm SKF/OOSM, without having to use an augmented state consisting of the target state and the sensor biases, which can become prohibitive for real-time implementation. This method was presented in the context of a Kalman filter and has also been extended to an IMM estimator. The SKF/OOSM algorithm was compared with the plain Kalman filter without compensation and the (heuristic) Kalman filter with covariance inflation, in the presence of residual biases. The simulation re-

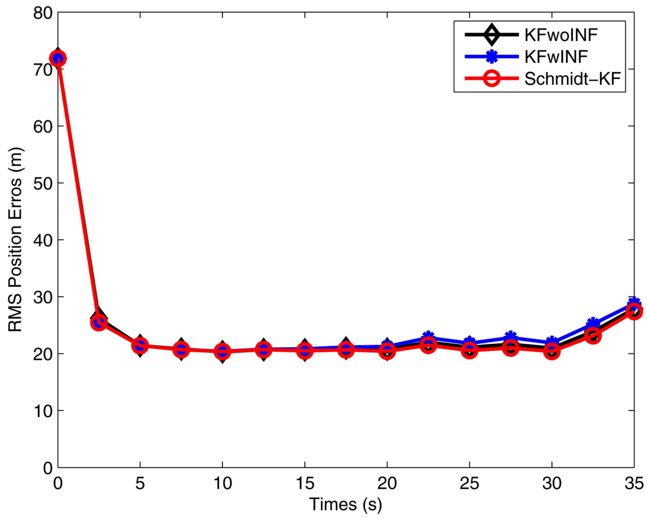


Fig. 17. Position RMSE for maneuvering target with GMTI measurement (small bias, reordering measurements).

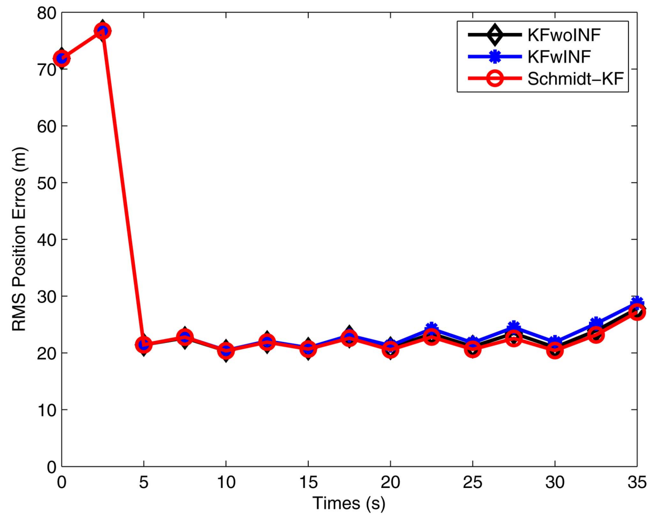


Fig. 19. Position RMSE for maneuvering target with GMTI measurement (small bias, OOSM processing).

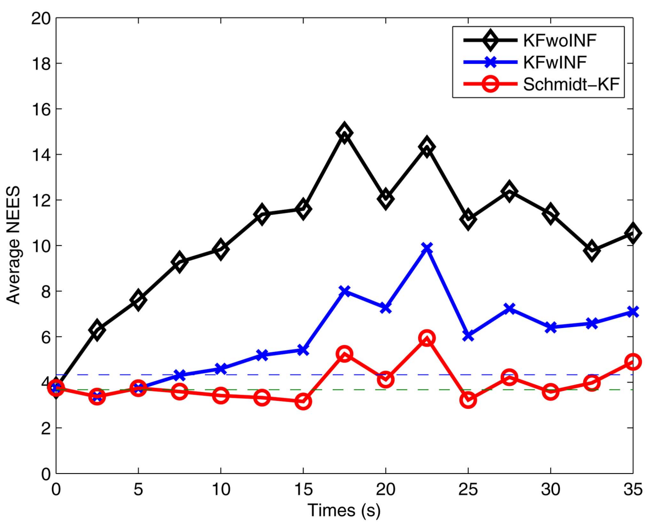


Fig. 18. NEES for maneuvering target with GMTI measurement (small bias, reordering measurements).

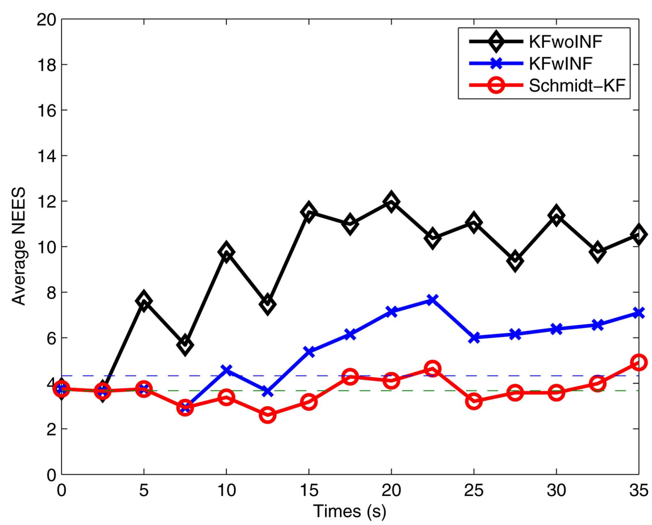


Fig. 20. NEES for maneuvering target with GMTI measurement (small bias, OOSM processing).

sults show that, compared to the other two methods, the major benefit of the SKF/OOSM algorithm is the significant improvement in filter consistency for both in-sequence data and processing OOSMs. For the estimation error, in the case using position only measurements, neither the SKF/OOSM algorithm nor the covariance-inflation method provide improvement in estimation accuracy over the plain Kalman filter without compensation. However, when GMTI measurements are used, which include additional range rate measurements, the SKF/OOSM algorithm outperforms the other two methods in both estimation accuracy and filter consistency.

APPENDIX A. DERIVATIONS OF JACOBIAN FOR GMTI MEASUREMENTS

As in (120), the GMTI measurement model is

$$z = (I_3 + \Lambda)h(\mathbf{x} - \mathbf{x}_p) + \Delta + w \tag{125}$$

where the superscript is dropped for simplicity and h is as in (122)

$$h(\mathbf{x}) = \begin{bmatrix} r \\ \theta \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \tan^{-1}\dfrac{y}{x} \\ \dot{x}\cos\theta + \dot{y}\sin\theta \end{bmatrix}. \tag{126}$$

In the sequel, the bias vector $\mathbf{b}$, which consists of the elements of Δ (the offset biases) as well as the diagonal elements of Λ (the multiplicative biases), is defined as

$$\mathbf{b} = [\mathbf{b}'_\Delta \quad \mathbf{b}'_\Lambda]' \tag{127}$$

where

$$\mathbf{b}_\Delta = \Delta, \qquad \mathbf{b}_\Lambda = [\alpha_r \quad \alpha_\theta \quad \alpha_{\dot{r}}]'. \tag{128}$$

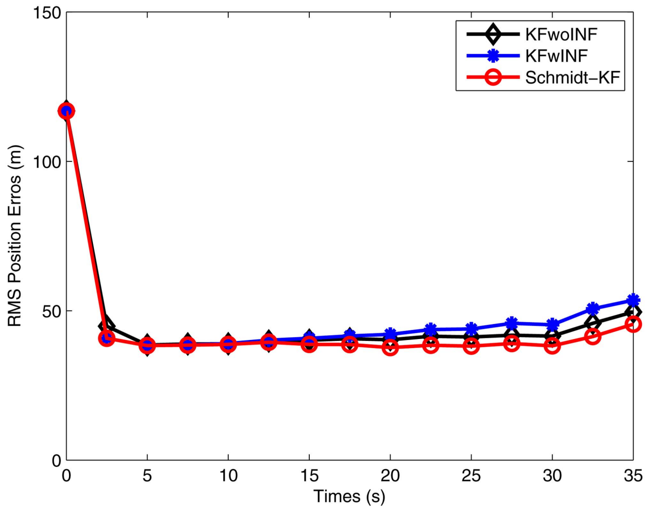


Fig. 21. Position RMSE for maneuvering target with GMTI measurement (large bias, reordering measurements).

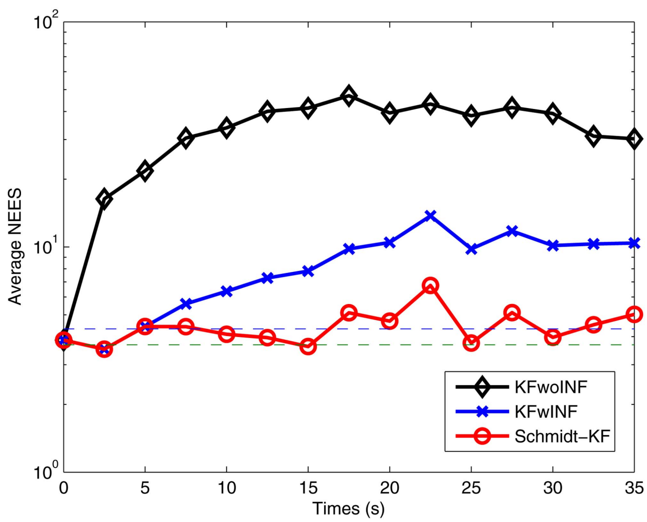


Fig. 22. NEES for maneuvering target with GMTI measurement (large bias, reordering measurements).

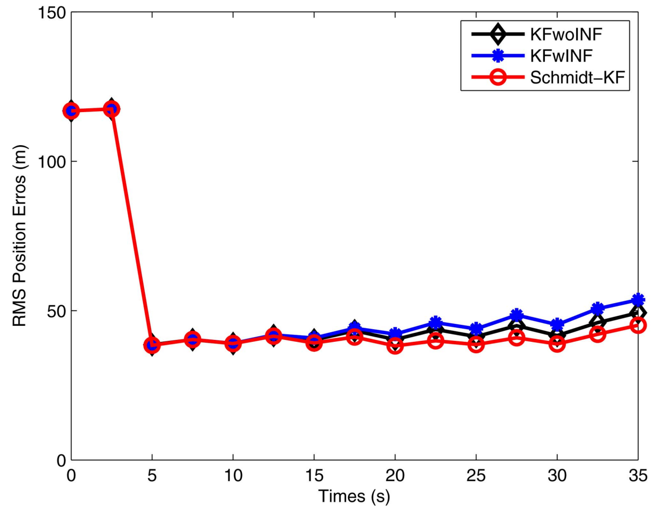


Fig. 23. Position RMSE for maneuvering target with GMTI measurement (large bias, OOSM processing).

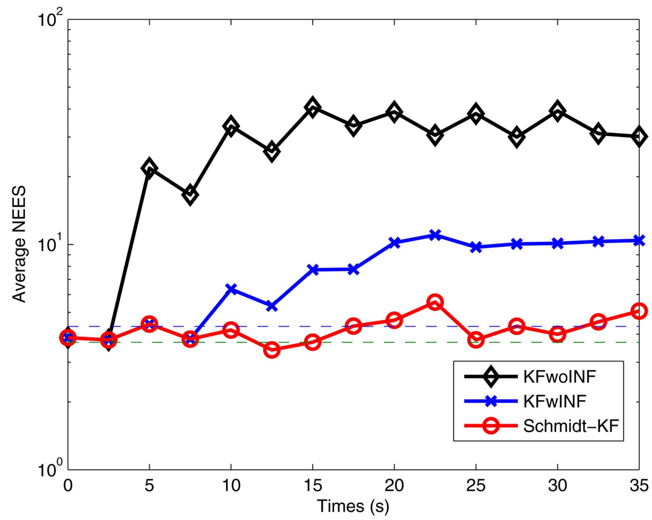


Fig. 24. NEES for maneuvering target with GMTI measurement (large bias, OOSM processing).

The Jacobian with respect to the target state, $H_{\mathbf{x}}$, is

$$\begin{aligned} H_{\mathbf{x}} &= \left.\frac{\partial z}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}},\mathbf{b}=\mathbf{0}} \\ &= (I_3+\Lambda)\left.\frac{\partial h(\mathbf{x}-\mathbf{x}_p)}{\partial(\mathbf{x}-\mathbf{x}_p)}\frac{\partial(\mathbf{x}-\mathbf{x}_p)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}},\mathbf{b}=\mathbf{0}} \\ &= H(\hat{\mathbf{x}}-\mathbf{x}_p), \end{aligned} \tag{129}$$

where $H(\mathbf{x})$ is

$$\begin{aligned} H(\mathbf{x}) &= \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \\ &= \begin{bmatrix} \frac{x}{r} & 0 & \frac{y}{r} & 0 \\ -\frac{y}{r^2} & 0 & \frac{x}{r^2} & 0 \\ l\sin\theta & \cos\theta & -l\cos\theta & \sin\theta \end{bmatrix} \end{aligned} \tag{130}$$

and

$$l = \frac{\dot{x}\sin\theta - \dot{y}\cos\theta}{r}. \tag{131}$$

The value of $\hat{\mathbf{x}}$ is $\hat{\mathbf{x}}(k \mid k-1)$ for a normal update and is $\hat{\mathbf{x}}(\kappa \mid k)$ for OOSM.

The Jacobian with respect to the bias, $H_{\mathbf{b}}$ can be divided into two parts, that is,

$$H_{\mathbf{b}} = [H_{\mathbf{b}_\Delta} \quad H_{\mathbf{b}_\Lambda}] \tag{132}$$

where $H_{\mathbf{b}_\Delta}$ is the Jacobian with respect to the offset bias $\mathbf{b}_\Delta$ and $H_{\mathbf{b}_\Lambda}$ is the Jacobian with respect to the scale bias $\mathbf{b}_\Lambda$, which are given by

$$\begin{aligned} H_{\mathbf{b}_\Delta} &= \left.\frac{\partial z}{\partial \mathbf{b}_\Delta}\right|_{\mathbf{x}=\hat{\mathbf{x}},\mathbf{b}=\mathbf{0}} \\ &= \left.\frac{\partial \Delta}{\partial \mathbf{b}_\Delta}\right|_{\mathbf{x}=\hat{\mathbf{x}},\mathbf{b}=\mathbf{0}} \\ &= I_3 \end{aligned} \tag{133}$$

$$
\begin{aligned}
H_{\mathbf{b}_\Lambda} &= \left.\frac{\partial z}{\partial \mathbf{b}_\Lambda}\right|_{\mathbf{x}=\hat{\mathbf{x}},\mathbf{b}=\mathbf{0}} \\
&= \left.\frac{\partial \Lambda h(\mathbf{x}-\mathbf{x}_p)}{\partial \mathbf{b}_\Lambda}\right|_{\mathbf{x}=\hat{\mathbf{x}},\mathbf{b}=\mathbf{0}} \\
&= \mathrm{diag}[h(\hat{\mathbf{x}}-\mathbf{x}_p)] \qquad (134)
\end{aligned}
$$

where diag[**a**] is a square matrix with the elements of the vector **a** on its main diagonal and the other elements zero.

REFERENCES

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

[2] Y. Bar-Shalom
Update with out-of-sequence measurements in tracking: Exact solution.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 3 (July 2002), 769–778.

[3] Y. Bar-Shalom, H. Chen, and M. Mallick
One-step solution for the multistep out-of-sequence measurement problem in tracking.
IEEE Transactions on Aerospace and Electronic Systems, **AES-40**, 1 (Jan. 2004), 27–37.

[4] Y. Bar-Shalom and H. Chen
IMM estimator with out-of-sequence measurements.
IEEE Transactions on Aerospace and Electronic Systems, **AES-41**, 1 (Jan. 2005), 90–98.

[5] S. S. Blackman and R. Popoli
Design and Analysis of Modern Tracking Systems.
Artech House, 1999.

[6] R. D. Hilton, D. A. Martin, and W. D. Blair
Tracking with Time-Delayed Data in Multisensor Systems.
NSWCDD/TR-93/351, Aug. 1993, Dahlgren, VA.

[7] A. H. Jazwinski
Stochastic Processes and Filtering Theory.
Academic Press, 1970.

[8] W. Koch
On Accumulatd State Densities with Applications to out-of-sequence measurement processing.
Proceedings of the 12th International Conference on Information Fusion, Seattle, WA, July 2009.

[9] X. D. Lin, Y. Bar-Shalom, and T. Kirubarajan
Exact multisensor dynamic bias estimation with local tracks.
IEEE Transactions on Aerospace and Electronic Systems, **AES-40**, 2 (Apr. 2004), 576–590.

[10] M. Mallick, S. Coraluppi, and Y. Bar-Shalom
Comparison of out-of-sequence measurements algorithms in multi-platform target tracking.
Proceedings of 2001 International Conference on Information Fusion, Montreal, Canada, Aug. 2001.

[11] M. Mallick, S. Coraluppi, and C. Carthel
Advances in asynchronous and decentralized estimation.
Proceedings of 2001 IEEE Aerospace Conference, Big Sky, MT, Mar. 2001.

[12] M. Mallick and A. Marrs
Comparison of the KF and particle filter based out-of-sequence measurement filtering algorithms.
Proceedings of 2003 International Conference on Information Fusion, Cairns, Australia, July 8–11, 2003.

[13] M. Mallick and K. Zhang
Optimal multiple-lag out-of-sequence measurement algorithm based on generalized smoothing framework.
Proceedings of Signal and Data Processing of Small Targets, San Diego, CA, July 31–Aug. 4, 2005.

[14] M. Orton and A. Marrs
Particle filters for tracking with out-of-sequence measurements.
IEEE Transactions on Aerospace and Electronic Systems, **41**, 2 (2005), 693–702.

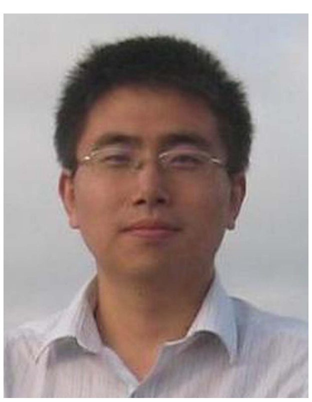
[15] R. Paffenroth, et al.
Mitigation of biases using the Schmidt-Kalman filter.
Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, vol. 6699, San Diego, CA, Aug. 2007.

[16] S. Schmidt
Applications of state space methods to navigation problems.
In C. T. Leondes (Ed.), *Advances in Control Systems*, vol. 3, Academic Press, 1966, 293–340.

[17] X. Shen, Y. Zhu, E. Song, and Y. Luo
Optimal centralized update with multiple local out-of-sequence measurements.
IEEE Transactions on Signal Processing, **57**, 4 (Apr. 2009), 1551–1562.

[18] G. Watson, D. H. McCabe, and T. Rice
Multisensor-multisite composite tracking in the presence of sensor residual bias.
Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, vol. 3809, Denver, CO, July 1999.

[19] K. Zhang, X. R. Li, and Y. Zhu
Optimal update with out-of-sequence measurements for distributed filtering.
IEEE Transactions on Signal Processing, **53**, 6 (June 2005), 1992–2004.



Shuo Zhang was born in 1982. He received the B.E. degree in 2005 and M.E. degree in 2007, both from the Department of Electrical Engineering, Harbin Institute of Technology, China.

He is now a Ph.D. candidate in the Department of Electrical and Computer Engineering, University of Connecticut. His research interests include statistical signal processing, tracking algorithms, and information fusion.

Yaakov Bar-Shalom (S'63—M'66—SM'80—F'84) was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967 and the Ph.D. degree from Princeton University, Princeton, NJ, in 1970, all in electrical engineering.

From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, CA. Currently he is Board of Trustees Distinguished Professor in the Department of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering. He is also director of the ESP Lab (Estimation and Signal Processing) at the University of Connecticut. His research interests are in estimation theory and stochastic adaptive control and he has published over 360 papers and book chapters in these areas. In view of the causality principle between the given name of a person (in this case, "(he) will track," in the modern version of the original language of the Bible) and the profession of this person, his interests have focused on tracking.

He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate text *Estimation with Applications to Tracking and Navigation* (Wiley, 2001), the text *Multitarget-Multisensor Tracking: Principles and Techniques* (YBS Publishing, 1995), and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Artech House, Vol. I 1990; Vol. II 1992, Vol. III 2000). He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He has been consulting to numerous companies, and originated the series of Multitarget Tracking and Multisensor Data Fusion short courses offered at Government Laboratories, private companies, and overseas.

During 1976 and 1977 he served as associate editor of the *IEEE Transactions on Automatic Control* and from 1978 to 1981 as associate editor of *Automatica*. He was program chairman of the 1982 American Control Conference, general chairman of the 1985 ACC, and cochairman of the 1989 IEEE International Conference on Control and Applications. During 1983–1987 he served as chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–1989 was a member of the Board of Governors of the IEEE CSS. Currently he is a member of the Board of Directors of the International Society of Information Fusion and served as its Y2K and Y2K2 President. In 1987 he received the IEEE CSS distinguished Member Award. Since 1995 he is a distinguished lecturer of the IEEE AESS. He is corecipient of the M. Barry Carlton Awards for the best paper in the *IEEE Transactions on Aerospace and Electronic Systems* in 1995 and 2000, and received the 1998 University of Connecticut AAUP Excellence Award for Research, the 2002 J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group, and the 2008 IEEE D. J. Picard Medal for Radar Technologies and Applications.

Gregory Watson biography not available.

Leg-by-leg Bearings-Only Target Motion Analysis Without Observer Maneuver

CLAUDE JAUFFRET

DENIS PILLON

ANNIE-CLAUDE PIGNOL

In a previous paper [7], the problem of bearings-only tracking of targets whose trajectory is composed of two legs from a non-maneuvering observer was addressed and the maximum likelihood estimate (MLE) proposed. We named it bearings-only maneuvering target motion analysis (BOMTMA). Recently in [9], we proposed another estimate based on leg-by-leg tracking and compare its performance to the MLE. We give here the extended version of [9], together with some comparison between the conventional bearings-only target motion analysis (BOTMA) and the BOMTMA.

Manuscript received November 12, 2009; revised October 27, 2010; released for publication February 08, 2011.

Refereeing of this contribution was handled by Huimin Chen.

Authors' addresses: C. Jauffret, Université Sud Toulon-Var, CNRS, IM2NP (UMR 6242), Bâtiment X, BP 132, 83957 La Garde Cedex, France, Email: (jauffret@univ-tln.fr); D. Pillon, Thales Underwater Systems, 525 Route des Dolines, 06903 Sophia-Antipolis, France, Email: (denis.pillon@fr.thalesgroup.com); A. C. Pignol, Université Sud Toulon-Var, CNRS, IM2NP (UMR 6242), Bâtiment X, BP 132, 83957 La Garde Cedex, France, Email: (pignol@univ-tln.fr).

1557-6418/11/$17.00 © 2011 JAIF

1. INTRODUCTION

The conventional problem of bearings-only target motion analysis consists of estimating the trajectory of a target (or source) whose velocity is constant during the period of measurement [13]. This requires an efficient maneuver of the observer to guarantee observability [6][12] and to obtain an accurate estimate [11][15]. In a recent paper [7], we proved that, conversely, if the observer has a constant velocity and the source changes its heading (so its trajectory is composed of two legs at constant speed—see Fig. 5), then, subject to a condition on velocity vectors of the two mobiles, the source is observable. For this problem, called bearings-only maneuvering target motion analysis (BOMTMA), we proposed the maximum likelihood estimate (MLE) and compared its performance with the Cramér-Rao lower bound (CRLB), revealing that this estimate is relatively efficient. The major criticisms are

1) The operator must wait until the source has changed its heading to run the computation of the estimate.
2) The computation by a numerical routine needs a "good guess" (to reduce the risk of converging toward a local minimum).
3) The computation takes time.

In this paper, we propose a new approach to this problem which consists of estimating what is observable during the first leg of the source, then during the second one, and finally of fusing these two estimated state vectors to obtain an estimate of the source trajectory. Note that in the classic BOTMA, the leg-by-leg approach has been employed for the same reasons [2][14][16]. We will assume that the maneuver time is known, hence we will not address the problem of detecting the maneuver (see [5] and [17] for this topic).

The paper is composed of three main sections:

- In Section 2, we present the problem of target motion analysis (TMA) when neither the source nor the observer maneuvers.
- A new bearings-only maneuvering target motion analysis by a non-maneuvering observer is proposed in Section 3.
- In Section 4, some examples are provided to compare respective performances of BOTMA and BOMTMA, in terms of estimated range accuracy.

2. PROBLEM FORMULATION WHEN NEITHER THE SOURCE NOR THE OBSERVER MANEUVERS

We consider in this section the case where the source and the observer are moving in the same plane with their own constant velocity vectors (see Fig. 1).

2.1. Measurement Equation and Trajectory Model

Consider a source and a passive observer (also called own ship). From here on, the subscript S is used to

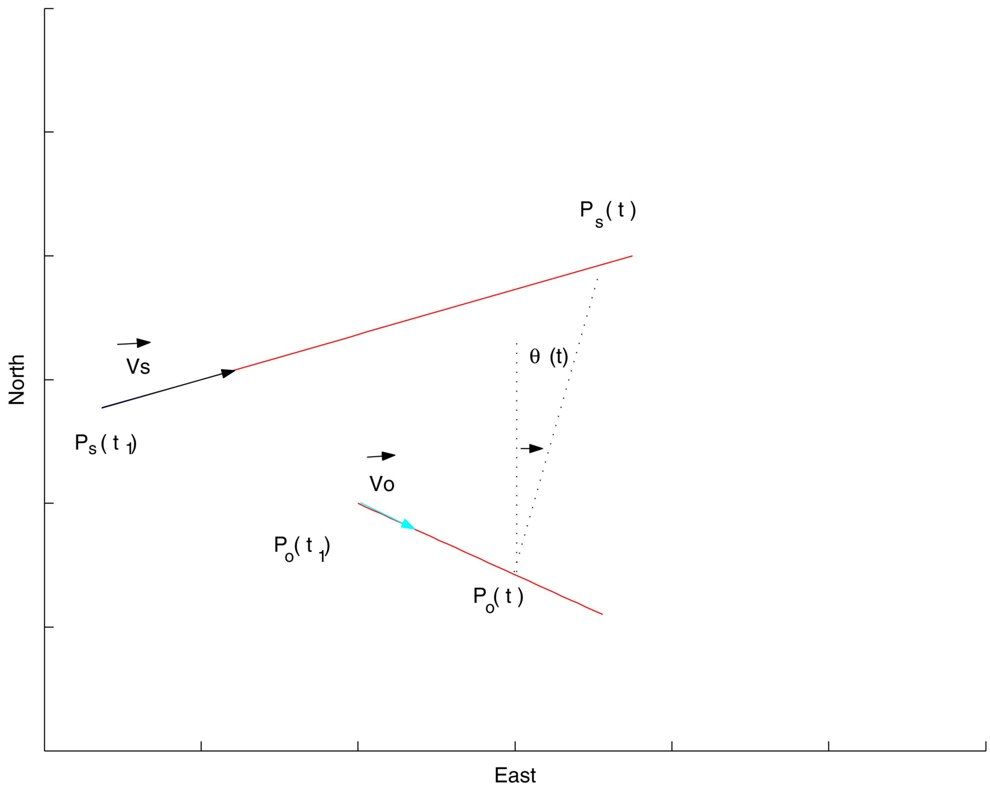


Fig. 1. Example of observer and source trajectories.

represent source quantities and O to represent observer quantities.

At time t, the respective location vectors of the source and of the observer are $P_S(t) = [x_S(t) \;\; y_S(t)]^{\mathrm{T}}$ and $P_O(t) = [x_O(t) \;\; y_O(t)]^{\mathrm{T}}$ relative to a Cartesian coordinate system. Similarly, the vectors $V_S = [\dot{x}_S \;\; \dot{y}_S]^{\mathrm{T}}$ and $V_O = [\dot{x}_O \;\; \dot{y}_O]^{\mathrm{T}}$ denote the source and own ship velocity vectors, respectively. We define also the relative velocity vector of the source w.r.t. the observer as $V_R = V_S - V_O$. The corresponding speeds and headings (or courses) are denoted v_S, v_O, v_R, c_S, c_O and c_R.

At time t_k, the observer measures the azimuth of the line of sight in which it detects the source:

$$\beta_k = \mathrm{atan}\left[\frac{x_S(t_k) - x_O(t_k)}{y_S(t_k) - y_O(t_k)}\right] + \varepsilon_k \qquad (1)$$

where ε_k is assumed to be a zero-mean Gaussian random noise of variance σ_k^2.

The BOTMA aims to estimate

$$X_S = [x_S(t^*) \;\; y_S(t^*) \;\; \dot{x}_S \;\; \dot{y}_S]^{\mathrm{T}}$$

from the collected measurement set $\{\beta_1, \beta_2, \ldots, \beta_N\}$ provided observability is guaranteed (t^* is an arbitrary reference time). The vector $X_O = [x_O(t^*) \;\; y_O(t^*) \;\; \dot{x}_O \;\; \dot{y}_O]^{\mathrm{T}}$ is known. It is well known that if the observer does not maneuver or if its maneuver is ambiguous (see [6] and [10]), then the vector X_S is not observable. In the coming paragraph, we explore the situations where the observer keeps its velocity vector during the scenario.

2.2. The Set of Homothetic Trajectories

The trajectory of a vehicle moving at a constant velocity vector $[\dot{x} \;\; \dot{y}]^{\mathrm{T}}$ is described by the following classic equations:

$$\begin{aligned} x(t) &= x(t^*) + (t - t^*)\dot{x} \\ y(t) &= y(t^*) + (t - t^*)\dot{y}. \end{aligned} \qquad (2)$$

Such a trajectory is hence defined by the vector $X = [x(t^*) \;\; y(t^*) \;\; \dot{x} \;\; \dot{y}]^{\mathrm{T}}$.

The equation of the noise-free bearings being $\theta(t) = \mathrm{atan}[(x_S(t) - x_O(t))/(y_S(t) - y_O(t))]$, it is straightforward to check that the set of trajectories producing the same noise-free-bearings from the observer is

$$\Lambda = \{X(\lambda) = \lambda(X_S - X_O) + X_O, \text{ for } \lambda > 0\}.$$

If $\theta(t)$ is not constant, there is no other trajectory set that generates the same noise-free data when the source and observer are moving at constant velocity vectors [10]. From now on, we will assume that $\theta(t)$ is not constant.

Note that the vector

$$X(\lambda) = [x_1(\lambda) \quad x_2(\lambda) \quad x_3(\lambda) \quad x_4(\lambda)]^{\mathrm{T}}$$

defines a λ-homothetic trajectory (in particular, $X(1) = X_S$).

It follows that X_S is not observable from the bearing measured by the observer. In short, in this context, the BOTMA is impossible. We can however estimate a parameter (or a state vector) that characterizes Λ. We call the estimation of this parameter (or any equivalent parameter) partial bearing-only target motion analysis (see Subsection 2.4).

The previous set Λ can be characterized by any of its elements $X(\lambda)$. At any $X(\lambda)$ of Λ, there is a corresponding unique three dimensional vector $Y = [y_1 \;\; y_2 \;\; y_3]^{\mathrm{T}}$ defined by

$$\begin{aligned} y_1 &= \operatorname{atan}\left[\frac{x_1(\lambda) - x_O(t^*)}{x_2(\lambda) - y_O(t^*)}\right] \\ y_2 &= \frac{\sqrt{(x_3(\lambda) - \dot{x}_O)^2 + (x_4(\lambda) - \dot{y}_O)^2}}{\sqrt{[x_1(\lambda) - x_O(t^*)]^2 + [x_2(\lambda) - y_O(t^*)]^2}} \\ y_3 &= \operatorname{atan}\left[\frac{x_3(\lambda) - \dot{x}_O}{x_4(\lambda) - \dot{y}_O}\right]. \end{aligned} \tag{3}$$

Indeed, the coordinates of Y are independent of λ:

$$y_1 = \theta(t^*), \qquad y_2 = \frac{v_R}{\rho_a(t^*)} \qquad \text{and} \qquad y_3 = c_R$$

where $\rho_a(t^*)$ is the actual range between the source and the observer at time t^* (see [6, 7]). Note that because $\theta(t)$ is not constant, we have $V_S \neq V_O$, hence $y_2 > 0$.

We can plot the set of homothetic trajectories Λ using the graphs of the two functions

$$\rho \mapsto v(\rho) = \sqrt{(\rho y_2 \sin y_3 + \dot{x}_O)^2 + (\rho y_2 \cos y_3 + \dot{y}_O)^2} \tag{4a}$$

$$\rho \mapsto c(\rho) = \operatorname{atan}\left[\frac{\rho y_2 \sin y_3 + \dot{x}_O}{\rho y_2 \cos y_3 + \dot{y}_O}\right] \tag{4b}$$

where $[v(\rho) \;\; c(\rho)]^{\mathrm{T}}$ are the polar coordinates of the velocity vector of any source of Λ at a distance ρ (≥ 0) at time t^* (the corresponding λ is equal to $\rho/\rho_a(t^*)$). Note that $v(\rho_a(t^*)) = v_S$ and $c(\rho_a(t^*)) = c_S$.

We insist on the fact that

1) any element of Λ allows us to construct the vector Y (see Eq. (3));

2) conversely, the vector Y allows us to construct any element of Λ, thank to the following equation

$$\begin{aligned} X\left(\frac{\rho}{\rho_a(t^*)}\right) &= \frac{\rho}{\rho_a(t^*)}(X_S - X_O) + X_O \\ &= \frac{\rho}{\rho_a(t^*)}\begin{bmatrix} \rho_a(t^*)\sin y_1 \\ \rho_a(t^*)\cos y_1 \\ \rho_a(t^*) y_2 \sin y_3 \\ \rho_a(t^*) y_2 \cos y_3 \end{bmatrix} + X_O. \end{aligned}$$

This equivalence between Y and Λ is the fundamental property of the partial bearings-only TMA which will be developed in Section 3. As a consequence, we can choose the vector Y as well any vector in Λ. The choice of a state vector must be guided by simplicity.

Because it is expressed in polar coordinates, Y is subject to a constraint: $y_2 > 0$, whereas any vector of Λ (expressed in Cartesian coordinates) is not. So, from the point of view of the estimation, choosing a particular vector of Λ as state vector is more convenient. A way to "stay" in Λ is to fix one coordinate of a 4-dimensional vector X and "adjust" the remained coordinates to Y:

For example, if we fix the first coordinate of an particular element of Λ to the value x_{fix}, the corresponding λ will be $\lambda = (x_{\mathrm{fix}} - x_O(t^*))/(x_S(t^*) - x_O(t^*))$; however, we must choose x_{fix} such that λ be positive. This will help us in Subsection 2.4.

2.3. Properties of $v(\rho)$ and $c(\rho)$

First of all, we note that $v(0)$ and $c(0)$ are equal to the observer's speed and heading, respectively. This corresponds to the degenerate case where the observer and the source are located at the same position.

2.3.1. Study of $v_S(\rho)$

Let us compute its derivative w.r.t. ρ.

$$\begin{aligned} \frac{d}{d\rho}v(\rho) &= \frac{1}{v(\rho)}[(\rho y_2 \sin y_3 + \dot{x}_O) y_2 \sin y_3 \\ &\qquad + (\rho y_2 \cos y_3 + \dot{y}_O) y_2 \cos y_3] \\ &= \frac{y_2}{v(\rho)}[\rho y_2 + \dot{x}_O \sin y_3 + \dot{y}_O \cos y_3] \\ &= \frac{y_2}{v(\rho)}\left[\rho y_2 + \frac{1}{v_R} V_O^{\mathrm{T}}(V_S - V_O)\right]. \end{aligned}$$

The sign of this derivative is hence the sign of $\rho y_2 + \dot{x}_O \sin y_3 + \dot{y}_O \cos y_3$. It is equal to 0 when $\rho = -(1/y_2 v_R) \cdot V_O^{\mathrm{T}}(V_S - V_O)$.

If $V_O^{\mathrm{T}}(V_S - V_O) \geq 0$ or equivalently $V_O^{\mathrm{T}} V_S \geq v_O^2$, the function $v_S(\rho)$ is injective, i.e. the mapping $\rho \mapsto v(\rho)$ satisfies the one-to-one condition.

We draw two other conclusions:

1) The one-to-one condition holds if and only if $\dot{x}_S \sin c_O + \dot{y}_S \cos c_O \geq v_O$.

2) The set of source's velocity vectors V_S satisfying the one-to-one condition of the speed is

$$\left\{ V_S = \alpha \begin{bmatrix} \sin c_O \\ \cos c_O \end{bmatrix} - \beta \begin{bmatrix} \cos c_O \\ \sin c_O \end{bmatrix}, \quad \text{for any } \alpha \geq v_O \text{ (no condition for } \beta\text{) and } V_S \neq V_O \right\}$$

Note that α and β that help define the above set are dummy variables.

2.3.2. Study of $c(\rho)$

First of all, note that $c(\rho)$ goes to c_R when $\rho \to \infty$.

A basic computation yields

$$\frac{d}{d\rho}c(\rho) = \frac{y_2(\dot{y}_O \sin y_3 - \dot{x}_O \cos y_3)}{v^2(\rho)}$$

which has the same sign as

$$v_R(\dot{y}_O \sin y_3 - \dot{x}_O \cos y_3) = -\det[V_O, (V_S - V_O)],$$

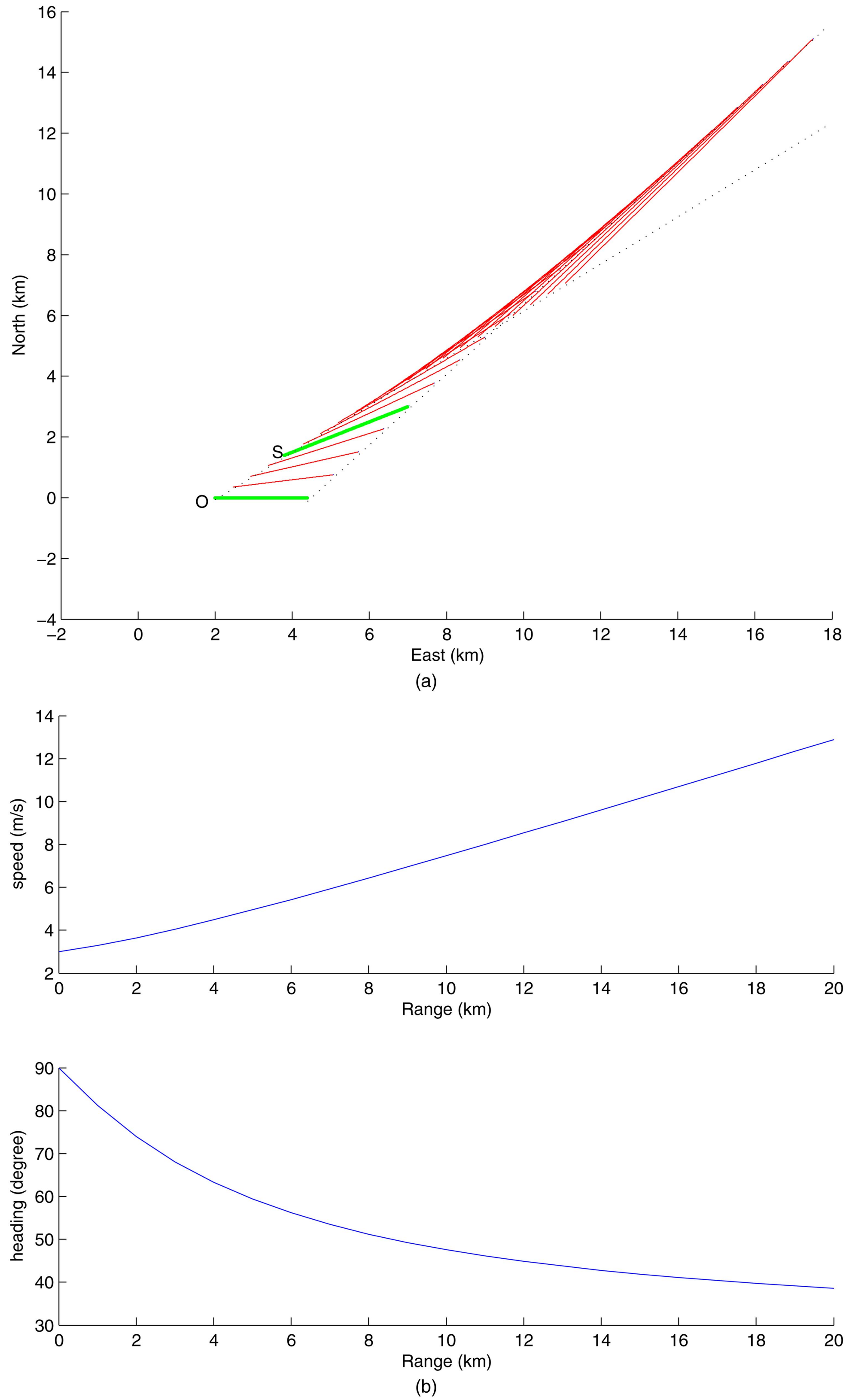


Fig. 2. (a) Some elements of Λ. (b) The corresponding speed and heading graphs.

which is independent of ρ. Hence the mapping $\rho \mapsto c(\rho)$ is monotonic while the mapping $\rho \mapsto v(\rho)$ can be not.

Fig. 2 gives an example of an increasing speed function for $V_O = [3\ \ 0]^{\mathrm{T}}$ (m/s) and $V_S = [4\ \ 2]^{\mathrm{T}}$ (m/s) corresponding to $(\alpha,\beta) = (4,2)$; the initial positions are $P_S(t_0) = [3.8\ \ 1.4]^{\mathrm{T}}$ (km) and $P_O(t_0) = [2\ \ 0]^{\mathrm{T}}$ (km). Note that the condition $\dot{x}_S \sin c_O + \dot{y}_S \cos c_O \geq v_O$ is satisfied. In Fig. 2(a), letters O and S denote the initial position

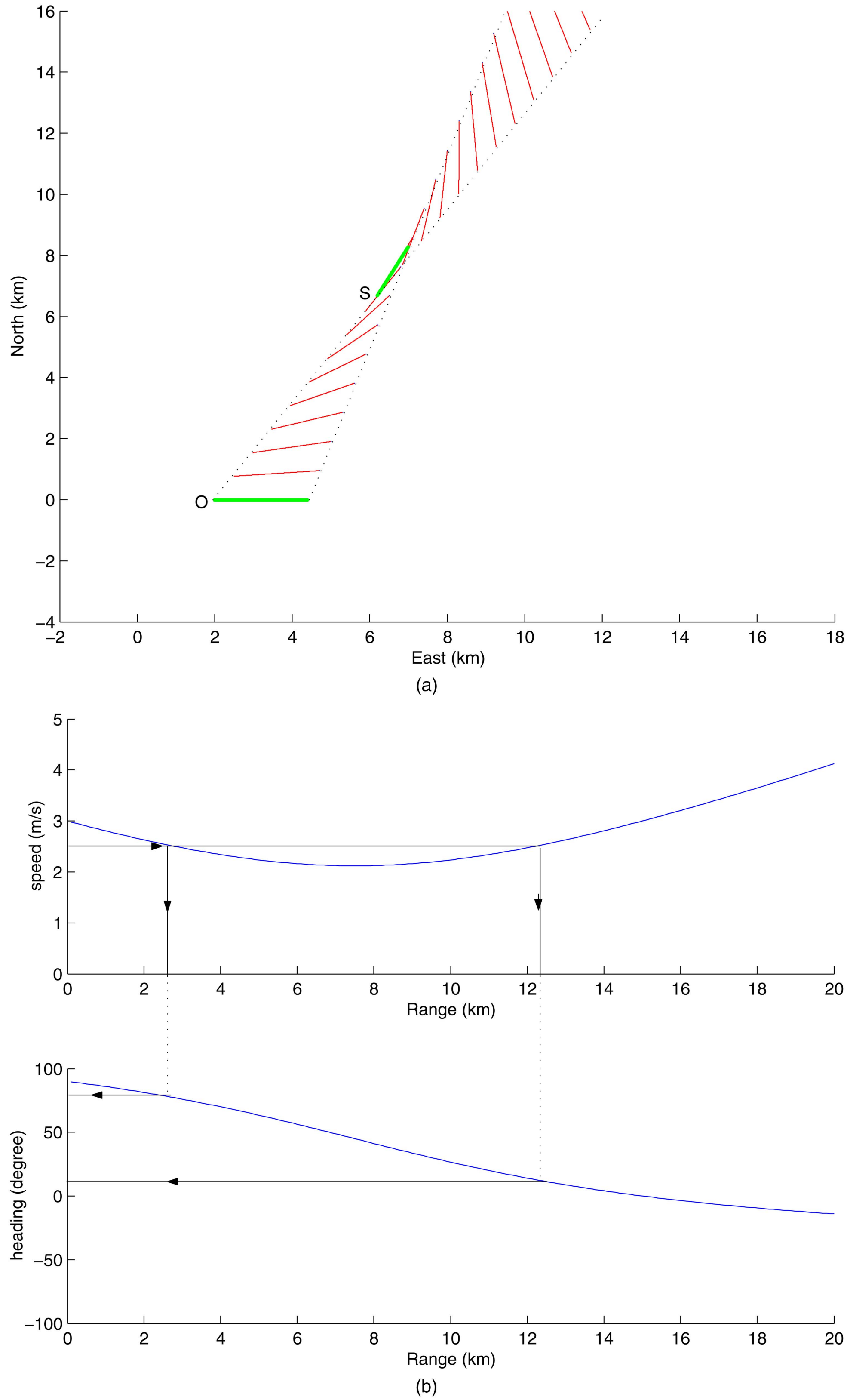


Fig. 3. (a) Some elements of Λ. (b) The corresponding speed and the heading graphs.

of the observer and that of the actual source and several homothetic solutions. Fig. 2(b) depicts $\rho \mapsto v(\rho)$ and $\rho \mapsto c(\rho)$; the small circles correspond to the actual speed and heading.

Fig. 3 illustrates the case of a non-monotonic speed function for $V_O = [3\ \ 0]^{\mathrm{T}}$ (m/s) and $V_S = [1\ \ 2]^{\mathrm{T}}$ (m/s) corresponding to $(\alpha, \beta) = (1, 2)$. The initial positions are $P_S(t_0) = [6.2\ \ 6.7]^{\mathrm{T}}$ (km) and $P_O(t_0) = [2\ \ 0]^{\mathrm{T}}$ (km). Here, the condition $\dot{x}_S \sin c_O + \dot{y}_S \cos c_O \geq v_O$ is violated.

In this case, due to the non-monotony of the mapping $\rho \mapsto v(\rho)$, a speed v would yield two corresponding estimated ranges and courses as shown in Fig. 3(b).

2.4. Estimation of the Set of Homothetic Trajectories (partial bearings-only TMA)

The available measured bearings $(\beta_1, \beta_2, \ldots, \beta_N)$ are taken at times $(t_1, t_2, \ldots, t_N)$. Without loss of generality, we choose $t^* = t_N$. Assuming the covariance matrix of the random vector $(\varepsilon_1 \ \ \varepsilon_2 \cdots \varepsilon_N)^{\mathrm{T}}$ to be diagonal, the log-likelihood function to be maximized is proportional to the least squares criterion. The vector Y or any vector in Λ can be chosen as state vector. Because of the simplicity of use of Cartesian coordinates, we propose to estimate a particular element of Λ by fixing its first coordinate x_1 to $\bar{\rho}\sin\beta_N + x_O(t_N)$ for convenience, $\bar{\rho}$ being arbitrarily chosen. We call it X. So we only have to compute the last three coordinates of $X = [\bar{\rho}\sin\beta_N + x_O(t_N) \ \ y \ \ \dot{x} \ \ \dot{y}]^{\mathrm{T}}$ for which the criterion $C(X) = \sum_{k=1}^{N}(1/\sigma_k^2)[\beta_k - \theta_k(X)]^2$ is minimal.

The Gauss-Newton method [3] is used for the minimization, initialized at

$$X_{\text{init}} = \begin{bmatrix} \bar{\rho}\sin\beta_N + x_O(t_N) \\ \bar{\rho}\cos\beta_N + y_O(t_N) \\ \dfrac{\bar{\rho}\sin\beta_N + x_O(t_N) - \bar{\rho}\sin\beta_1 - x_O(t_1)}{t_N - t_1} \\ \dfrac{\bar{\rho}\cos\beta_N + x_O(t_N) - \bar{\rho}\cos\beta_1 - x_O(t_1)}{t_N - t_1} \end{bmatrix}.$$

As pointed out previously, the maximum likelihood estimate $\hat{X}$ allows us to construct the set of estimated homothetic trajectories presented as the graphs of the pair of functions

$$\rho \mapsto \hat{v}(\rho) = \sqrt{(\rho\hat{y}_2\sin\hat{y}_3 + \dot{x}_O)^2 + (\rho\hat{y}_2\cos\hat{y}_3 + \dot{y}_O)^2} \tag{5a}$$

$$\rho \mapsto \hat{c}(\rho) = \operatorname{atan}\left[\frac{\rho\hat{y}_2\sin\hat{y}_3 + \dot{x}_O}{\rho\hat{y}_2\cos\hat{y}_3 + \dot{y}_O}\right] \tag{5b}$$

where $\hat{Y} = [\hat{y}_1 \ \ \hat{y}_2 \ \ \hat{y}_3]^{\mathrm{T}}$ is the vector corresponding to $\hat{X}$ (with (3)).

The behavior of the estimator $\hat{X}$ has been evaluated for the following scenario: given a coordinate system, the initial location of the source is $[1000 \ \ 2300]^{\mathrm{T}}$ (m) with a velocity vector of $[1 \ \ 1.5]^{\mathrm{T}}$ (m/s). The observer starts from $[0 \ \ 0]^{\mathrm{T}}$ (m) with a velocity vector $[1 \ \ 0]^{\mathrm{T}}$ (m/s). The number of measurements is $N = 450$, the time t_k is equal to $k \times \Delta t$, with $\Delta t = 4$ s. The time of reference is chosen to be t_N. The standard deviation of the measurements is 1°. The final range is 5,099 m. For the initialization of the Gauss-Newton method, we have chosen $\bar{\rho} = 20$ km.

Fig. 4 depicts an example of a 500-run Monte-Carlo simulation: the 500 graphs are plotted in grey, while the graphs of the functions $v(\rho)$ and $c(\rho)$ are in black, together with the 95% confidence bands (deduced from the CRLB). Detail of the computation of these bands is given in the Appendix.

3. LEG-BY-LEG BOMTMA

3.1. Problem Formulation

Suppose now that the trajectory of the source is composed of two legs at constant speed (cf. Fig. 5): the first leg starts at t_1 and finishes at time t_M (assumed to be known). Similarly, the second leg starts at t_M and finishes at t_K. This model of trajectory is simple, but it has been widely adopted in the past, especially in submarine environment (see [1] pp. 175–176 and [4]). The time of the maneuver is assumed to be known; in reality, it has to be estimated, for example by a sequential test; this point, which is out of the scope of this paper, has been addressed in [8]. Such a trajectory is hence parameterized by the vector $Z = [x_S(t_K) \ \ y_S(t_K) \ \ v_S \ \ c_{S,1} \ \ c_{S,2}]^{\mathrm{T}}$ (coordinates of position at time t_K, speed, courses of the first and of second leg). Provided that $V_O^{\mathrm{T}}(V_{S,1} - V_{S,2}) \neq 0$ (observability condition-see its proof in [7]), the entire source trajectory is observable. For this problem, we proposed the maximum likelihood estimate in [7].

We propose here another estimate denoted $\tilde{Z}$ the principle of which is as follows: First, we compute, for leg #1, the estimate

$$\hat{X}_1 = [\bar{\rho}\sin\beta_M + x_O(t_M) \quad \hat{y}_1 \quad \hat{\dot{x}}_1 \quad \hat{\dot{y}}_1]^{\mathrm{T}}$$

and for leg #2, the estimate

$$\hat{X}_2 = [\bar{\rho}\sin\beta_M + x_O(t_M) \quad \hat{y}_2 \quad \hat{\dot{x}}_2 \quad \hat{\dot{y}}_2]^{\mathrm{T}}$$

with the common reference time t_M and after having fixed their respective first coordinates to a common value, say $\bar{\rho}\sin\beta_M + x_O(t_M)$. These estimates are computed following the partial BO-TMA principle as presented in Section 2.4. Second, we compute the two homothetic estimates, $\mu(\hat{X}_1 - X_O) + X_O$ for the first leg and $\mu(\hat{X}_2 - X_O) + X_O$ for the second, such that the estimated velocities on each leg are equal, i.e.

$$\|\mu(\hat{V}_1 - V_O) + V_O\| = \|\mu(\hat{V}_2 - V_O) + V_O\| \tag{6}$$

with $\hat{V}_1 = [\hat{\dot{x}}_1 \ \ \hat{\dot{y}}_1]^{\mathrm{T}}$ and $\hat{V}_2 = [\hat{\dot{x}}_2 \ \ \hat{\dot{y}}_2]^{\mathrm{T}}$ in order to satisfy the constraint (6) (which is the strong assumptions of the BOMTMA). The solution of (6), denoted $\tilde{\mu}$, and equal to

$$\tilde{\mu} = -\frac{2(\hat{V}_1 - \hat{V}_2)^{\mathrm{T}}V_O}{(\|\hat{V}_1\|^2 - \|\hat{V}_2\|^2 - 2(\hat{V}_1 - \hat{V}_2)^{\mathrm{T}}V_O)}$$

allows us to compute the corresponding homothetic estimates on each leg

$$\tilde{X}_1 = \tilde{\mu}(\hat{X}_1 - X_O) + X_O = [\tilde{x}_1 \quad \tilde{y}_1 \quad \tilde{\dot{x}}_1 \quad \tilde{\dot{y}}_1]^{\mathrm{T}}$$

$$\tilde{X}_2 = \tilde{\mu}(\hat{X}_2 - X_O) + X_O = [\tilde{x}_2 \quad \tilde{y}_2 \quad \tilde{\dot{x}}_2 \quad \tilde{\dot{y}}_2]$$

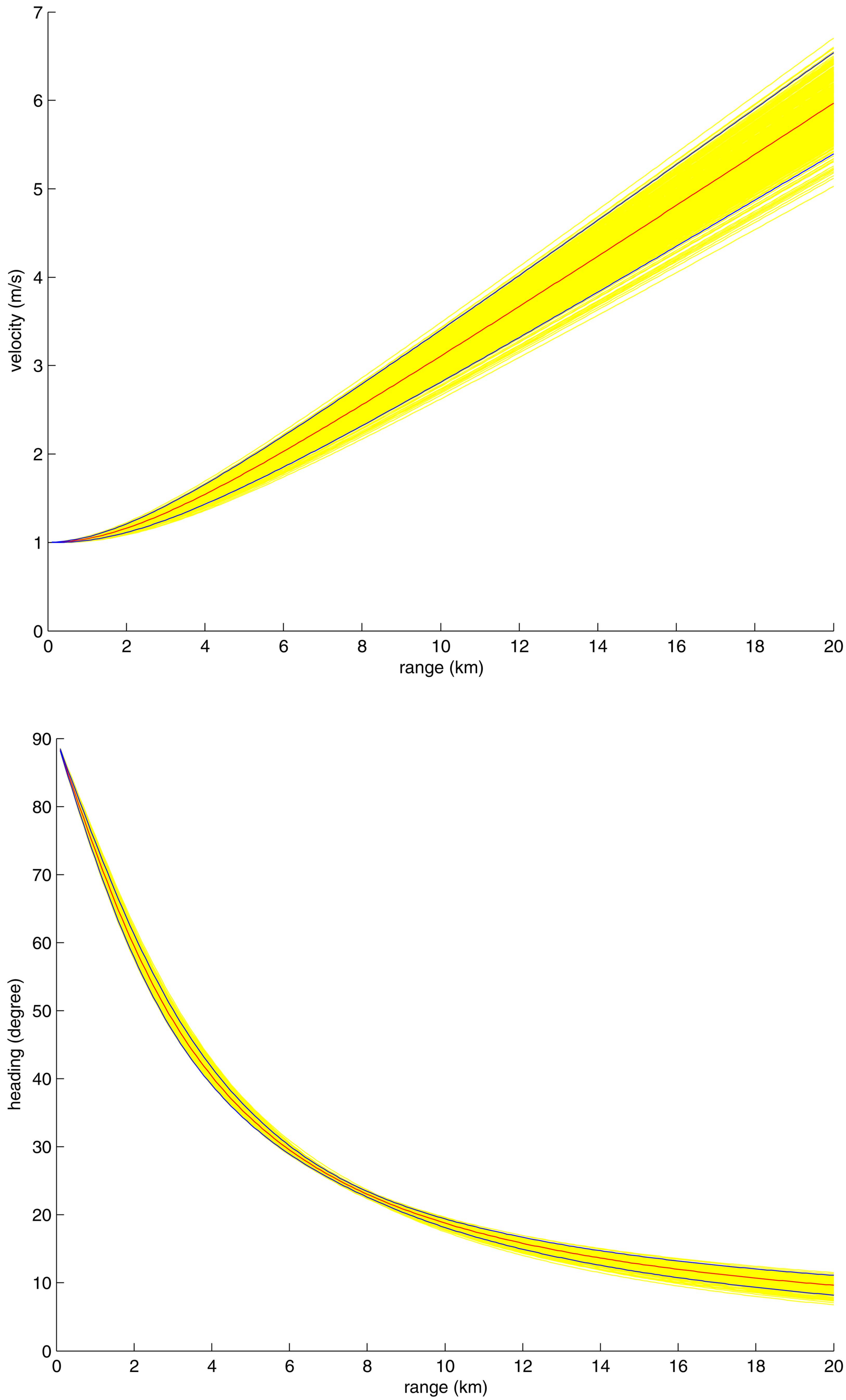


Fig. 4. Results of 500 Monte-Carlo runs (on left $\hat{v}(\rho)$, and on right $\hat{c}(\rho)$).

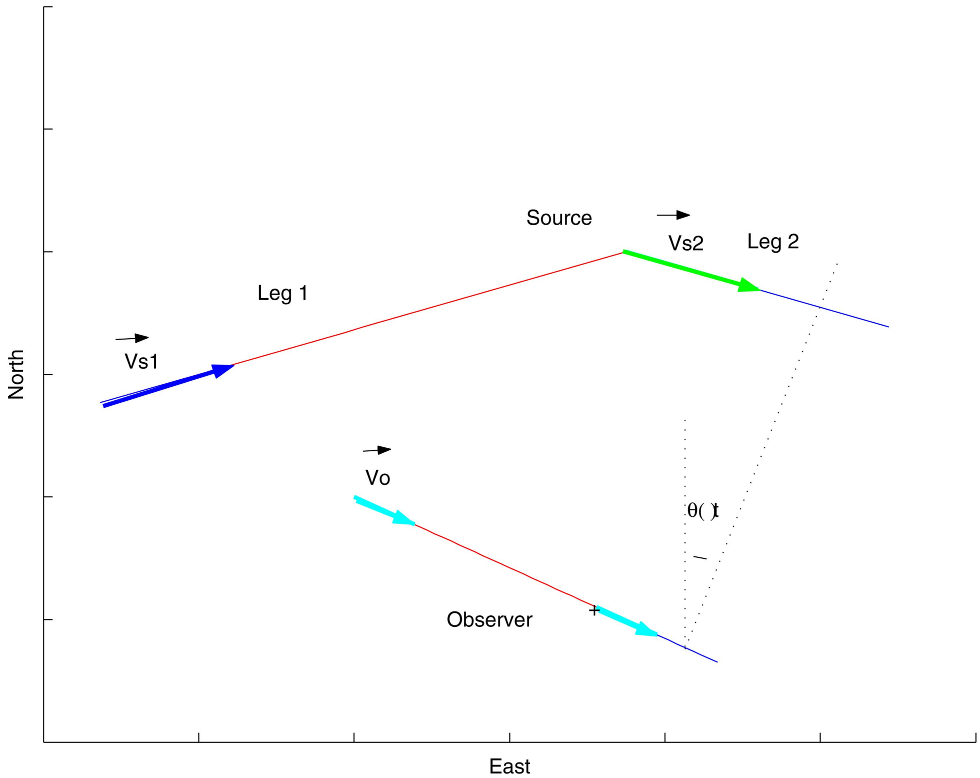


Fig. 5. Example of observer and source trajectories composed by two legs.

and the corresponding leg-by-leg BOMTMA $\tilde{Z} = [\tilde{x}_S(t_K)\ \tilde{y}_S(t_K)\ \tilde{v}_S\ \tilde{c}_{S,1}\ \tilde{c}_{S,2}]^{\mathrm{T}}$, with

$$[\tilde{x}_S(t_K) \quad \tilde{y}_S(t_K)]^{\mathrm{T}} = [\tilde{x}_2 \quad \tilde{y}_2]^{\mathrm{T}}$$

$$\tilde{v}_S = \sqrt{\tilde{\dot{x}}_2^2 + \tilde{\dot{y}}_2^2}$$

$$\tilde{c}_{S,1} = \operatorname{atan}\left(\frac{\tilde{\dot{x}}_1}{\tilde{\dot{y}}_1}\right)$$

$$\tilde{c}_{S,2} = \operatorname{atan}\left(\frac{\tilde{\dot{x}}_2}{\tilde{\dot{y}}_2}\right).$$

REMARK by construction, $\tilde{x}_1 = \tilde{x}_2$ and $\sqrt{\tilde{\dot{x}}_1^2 + \tilde{\dot{y}}_1^2} = \sqrt{\tilde{\dot{x}}_2^2 + \tilde{\dot{y}}_2^2}$, but there is no reason that $\tilde{y}_1 = \tilde{y}_2$. So, another solution must be $[\tilde{x}_S(t_K)\ \tilde{y}_S(t_K)]^{\mathrm{T}} = [\tilde{x}_2\ \tilde{y}_1]^{\mathrm{T}}$.

3.2. Problem Formulation

A 500-run Monte Carlo simulation allows the behavior of this new estimator to be appreciated. To compare the MLE BOMTMA estimate $\hat{Z}$ and the leg-by-leg BOMTMA estimate $\tilde{Z}$, we use the scenario presented in [7] which is illustrated in Fig. 6: let us recall that the observer starts from the origin with a speed of 5 m/s and a heading of 90°. Meanwhile, the source, with a speed of 4 m/s, starts its trajectory at $[0\ \mathrm{km}, 10\ \mathrm{km}]^{\mathrm{T}}$ with an initial heading of 90°. At time $t_M = 20$ min, it suddenly changes its course and its new heading is 240°. The total duration of the scenario is 30 min corresponding to 450 measurements (the sampling time is $\Delta t = 4$ s). The standard deviation of the measurement noise is 1°.

The average values of the coordinates of the two estimates, their respective biases and their empirical standard deviations are given in Table I. They are compared to the true vector and the minimum standard deviations deduced from the CRLB.

We observe an increase in the bias and the standard deviation, but the quality of the leg-by-leg BOMTMA estimator is only weakly degraded. Moreover, the computation time of the leg-by-leg BOMTMA estimator is 2.5 times less than the BOMTMA computation time. A compromise can probably be found: the leg-by-leg estimate can be used as an initial point for the BOMTMA numerical routine. It will reduce the risk of stalling at a local minimum. Fig. 7 shows the 500 estimates together with the 90% confidence ellipsoid deduced from the CRLB.

4. COMPARISON OF THE RESPECTIVE PERFORMANCES OF THE BOMTMA AND THE CONVENTIONAL BOTMA

One can ask a relevant question concerning tactical aspects: do situations exist in which the performance of the BOMTMA is superior to the performance of the BOTMA in terms of estimated range accuracy, assuming that the antennas are the same?

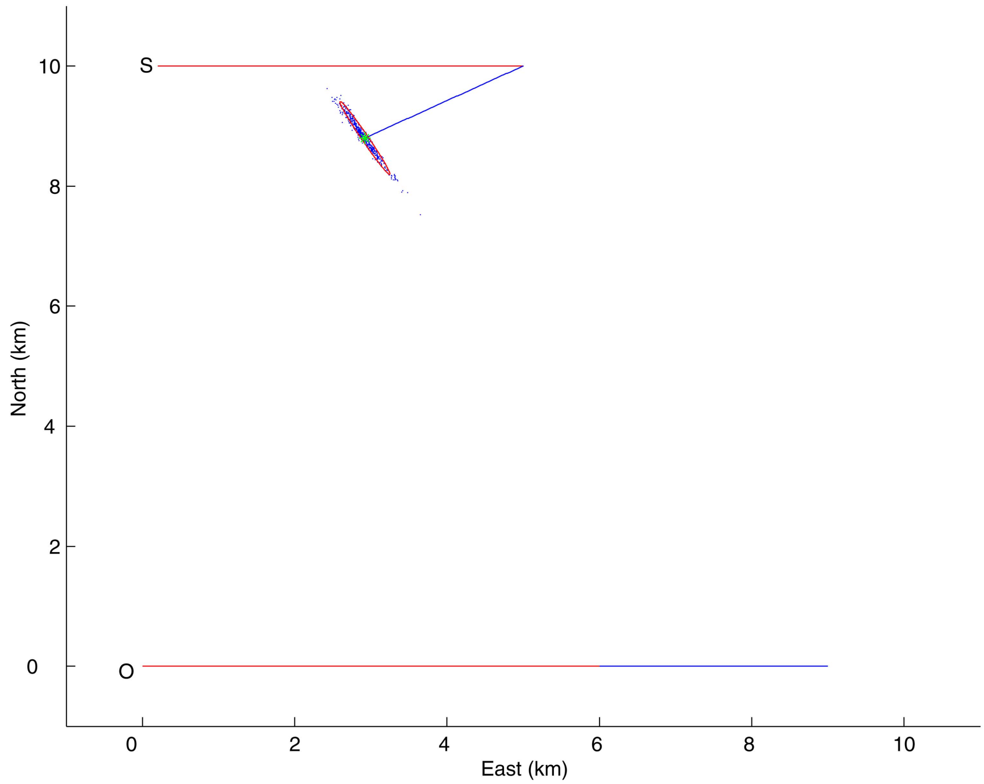


Fig. 6. 500 MLE BOMTMA estimates of the final position.

TABLE I
Comparison of Performances of the Two Estimators

Z_K	Units	$Z_{K,\text{True}}$	Average of $\hat{Z}_K$	Average of $\tilde{Z}_K$	Bias of $\hat{Z}$	Bias of $\tilde{Z}$	σ_{CRLB}	$\hat{\sigma}$ of $\hat{Z}$	$\tilde{\sigma}$ of $\tilde{Z}$
$x_S(t_K)$	km	2.921	2.897	**2.872**	0.024	**0.049**	0.153	0.175	**0.213**
$y_S(t_K)$	km	8.800	8.828	**8.875**	0.028	**0.075**	0.283	0.308	**0.364**
v_S	m/s	4	4.10	**4.10**	0.10	**0.10**	0.03	0.13	**0.17**
$c_{S,1}$	degree	90	90.5	**89.4**	0.5	**0.6**	12	12	**13**
$c_{S,2}$	degree	240	240	**241**	0	**1**	7.5	7	**9**

For the conventional BOTMA, the observer must correctly maneuver and it has to estimate a 4-dimensional state vector [13], whereas for the BOMTMA, the observer does not maneuver but it has to estimate a 5-dimensional vector. Because the number of unknown is less in BOTMA than in BOMTMA, one can think that the BOMTMA returns a less accurate estimated range than the BOTMA. Surprisingly, for some scenarios, this statement is wrong. We give three examples: the first one contradicts the intuition; for the second, the performances are similar, and the last is an example of the superiority of the BOTMA to the BOMTMA.

We consider two mobiles: one maneuvers (denoted hereafter by the letter "M") and the second (denoted "N") does not. Each of them performs a TMA against the other.

For each example, the speeds of M and N are 4 m/s and 5 m/s, respectively. The heading of the non-maneuvering platform is equal to 90 degrees. The kinds of maneuvering source trajectory have been chosen: a "surrounding" trajectory (see Fig. 8(a)) and two "escaping" trajectories (see Figs. 9(a) and 10(a)). The total duration is 30 min. The maneuvering platform changes its course at 15 min. At this time, its location is aligned with the course of the observer.

The bearings, collected at a sample time of 4 seconds by each platform, are corrupted by an additive Gaussian noise with the same standard deviation equal to 1 degree. We have computed the relative accuracy of the estimated range (at the final time t_K) by the mean of the CRLB: In $\sigma_{\rho(t_K)}/\rho(t_K)$, given in percentage, $\sigma_{\rho(t_K)}$ is the CRLB of the estimated final range.

Typical scenarios are plotted in Figs. 8, 9 and 10, ("M" and "N" give the initial positions of the two platforms) together with the corresponding $\sigma_{\rho(t_K)}/\rho(t_K)$ vs. $\rho(t_K)$. In these figures, the lines joining circles are related to the conventional BOTMA and the lines joining the "+" are for the BOMTMA. We have changed the final range by modifying the initial position of the non-maneuvering platform along the x-axis.

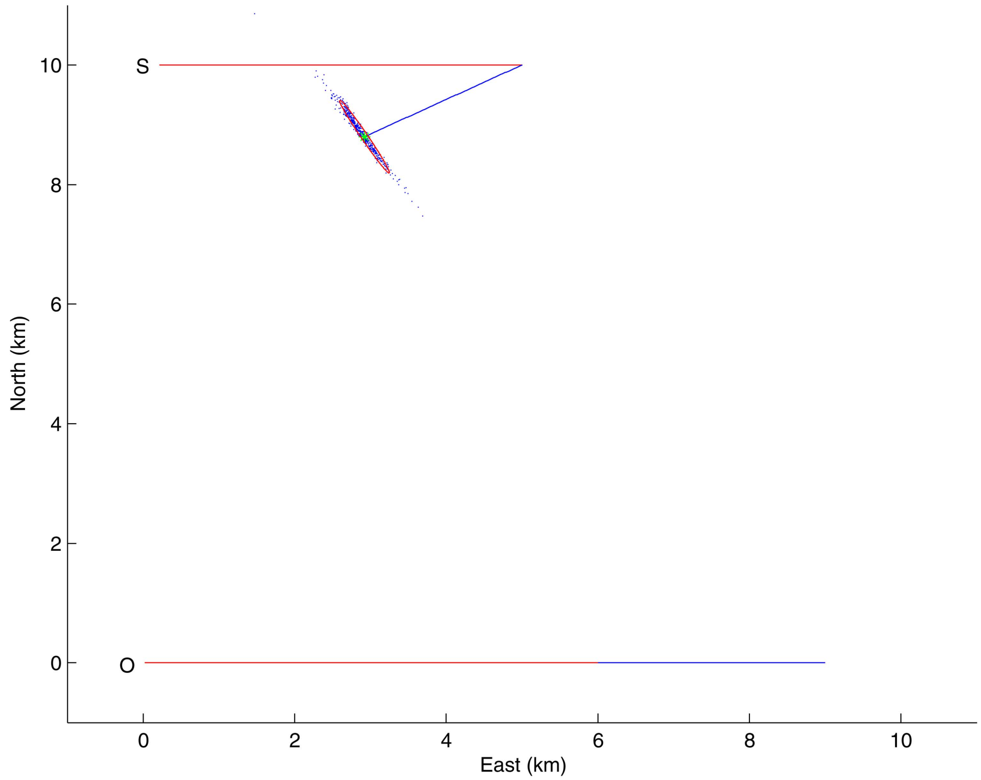


Fig. 7. 500 leg-by-leg BOMTMA estimates of the final position.

EXAMPLE 1 In this scenario, the first and the second headings of the maneuvering source are respectively $-135°$ and $135°$.

EXAMPLE 2 The first and the second headings of the maneuvering source are respectively $135°$ and $-135°$.

EXAMPLE 3 The first and the second headings of the maneuvering source are respectively $-135°$ and $135°$.

5. CONCLUSIONS

We have presented the problem of bearing-only target motion analysis when neither the source nor the observer maneuvers. This yields the so called partial BOTMA, since the observability is missing. Then, we have proposed another estimate for a two-leg source trajectory by bearings-only TMA from a non-maneuvering observer, which is an alternative solution of the maximum likelihood estimate proposed in [7]. The computation time is reduced by a factor of approximately 2.5. The price is a small degradation in the statistical performance. We have also shown, by three examples, that the superiority of the BOTMA to the BOMTMA is not always guaranteed.

Further work will be carried out to extend this estimation principle (leg-by-leg estimates, then fusion of them) to the case of several legs. Robustness to the assumption of constant speed and of the immediate change of heading will be the topic of another paper [8].

But the challenge remains to construct a powerful test to detect the maneuver of the source.

APPENDIX. COMPUTATION OF THE CONFIDENCE BANDS

To conduct properly the computation, we need to define the following vector $\tilde{X} = [y\ \ \dot{x}\ \ \dot{y}]^{\mathrm{T}}$ whose components are the last there components of X (defined in Subsection 2.4). We rename the components of $\tilde{X}$ as follows: $\tilde{X} = [x_1\ \ x_2\ \ x_3]^{\mathrm{T}}$ for convenience and we define the function

$$\theta_k(\tilde{X}) = \mathrm{atan}\left[\frac{\bar{\rho}\sin\beta_{t^*} + (t_k - t^*)(x_2 - \dot{x}_O)}{x_1 - y_O(t^*) + (t_k - t^*)(x_3 - \dot{y}_O)}\right]$$

which would be the noise-free bearing of a source at time t_k, whose trajectory would be defined by X.

Under classic Gaussian assumption about the additive noise, the Fisher information matrix is

$$F(\tilde{X}) = \sum_{k=1}^{K} \frac{1}{\sigma_k^2} \nabla_{\tilde{X}}\theta_k(\tilde{X})\nabla_{\tilde{X}}^{\mathrm{T}}\theta_k(\tilde{X})$$

and the CRLB of $\tilde{X}$ is $B(\tilde{X}) = F(\tilde{X})^{-1}$.

We go into detail about the expression of $\nabla_{\tilde{X}}\theta_k(\tilde{X})$ by defining the following quantities

$$\Delta_t(t_k) = t_k - t^*$$

$$\Delta_x(t_k) = \bar{\rho}\sin\beta_{t^*} + \Delta_t(t_k)(x_2 - \dot{x}_O),$$

$$\Delta_y(t_k) = x_1 - y_O(t^*) + \Delta_t(t_k)(x_3 - \dot{y}_O),$$

$$r(t_k) = \sqrt{\Delta_x^2(t_k) + \Delta_y^2(t_k)}.$$

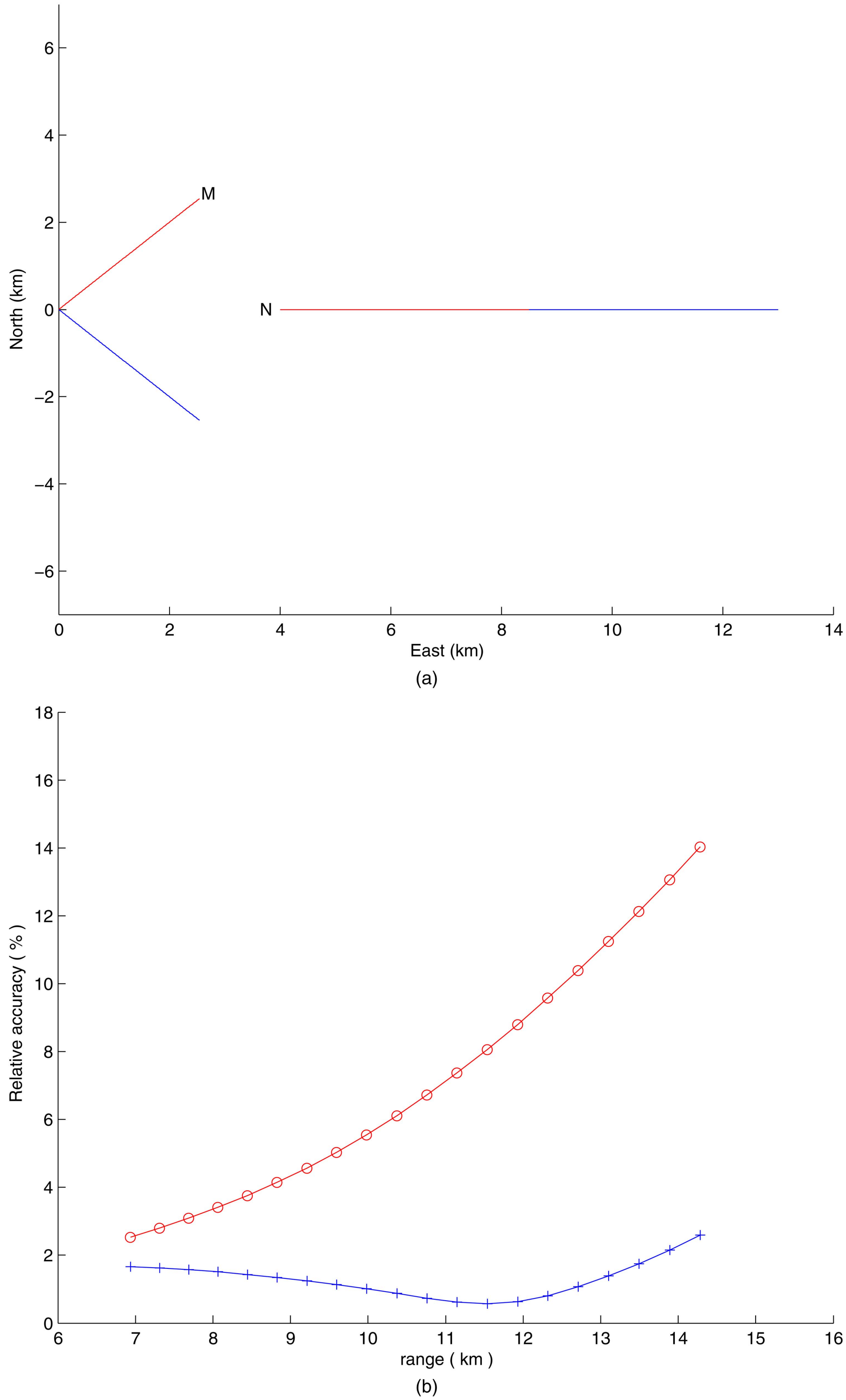


Fig. 8. (a) Typical surrounding maneuvering platform trajectory for Example 1. (b) The relative accuracies of the estimated ranges for the first example ("o" for the conventional BOTMA, "+" for the BOMTMA).

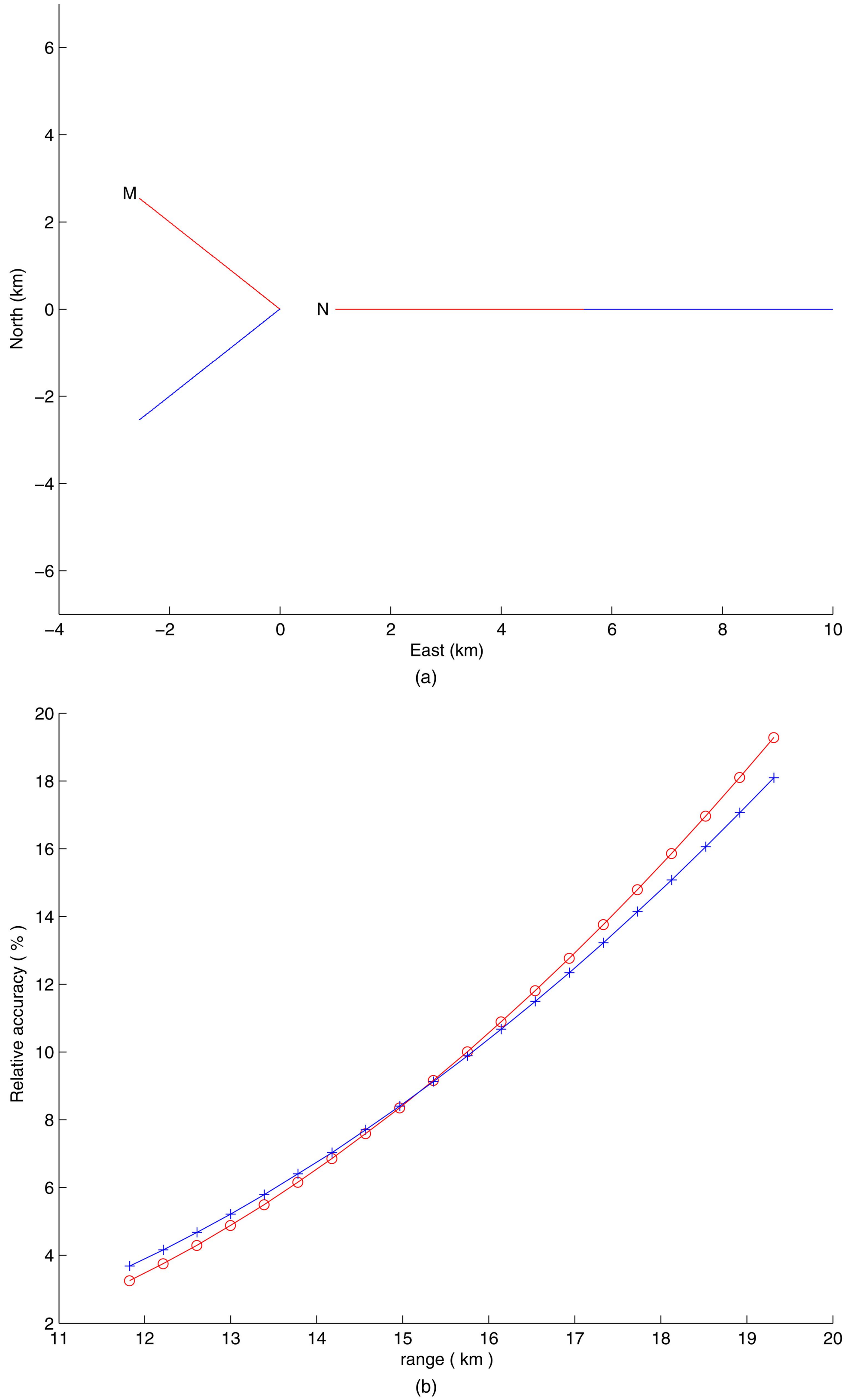


Fig. 9. (a) Typical escaping maneuvering platform for the second example. (b) The relative accuracies of the estimated ranges for the second example ("o" for the conventional BOTMA, "+" for the BOMTMA).

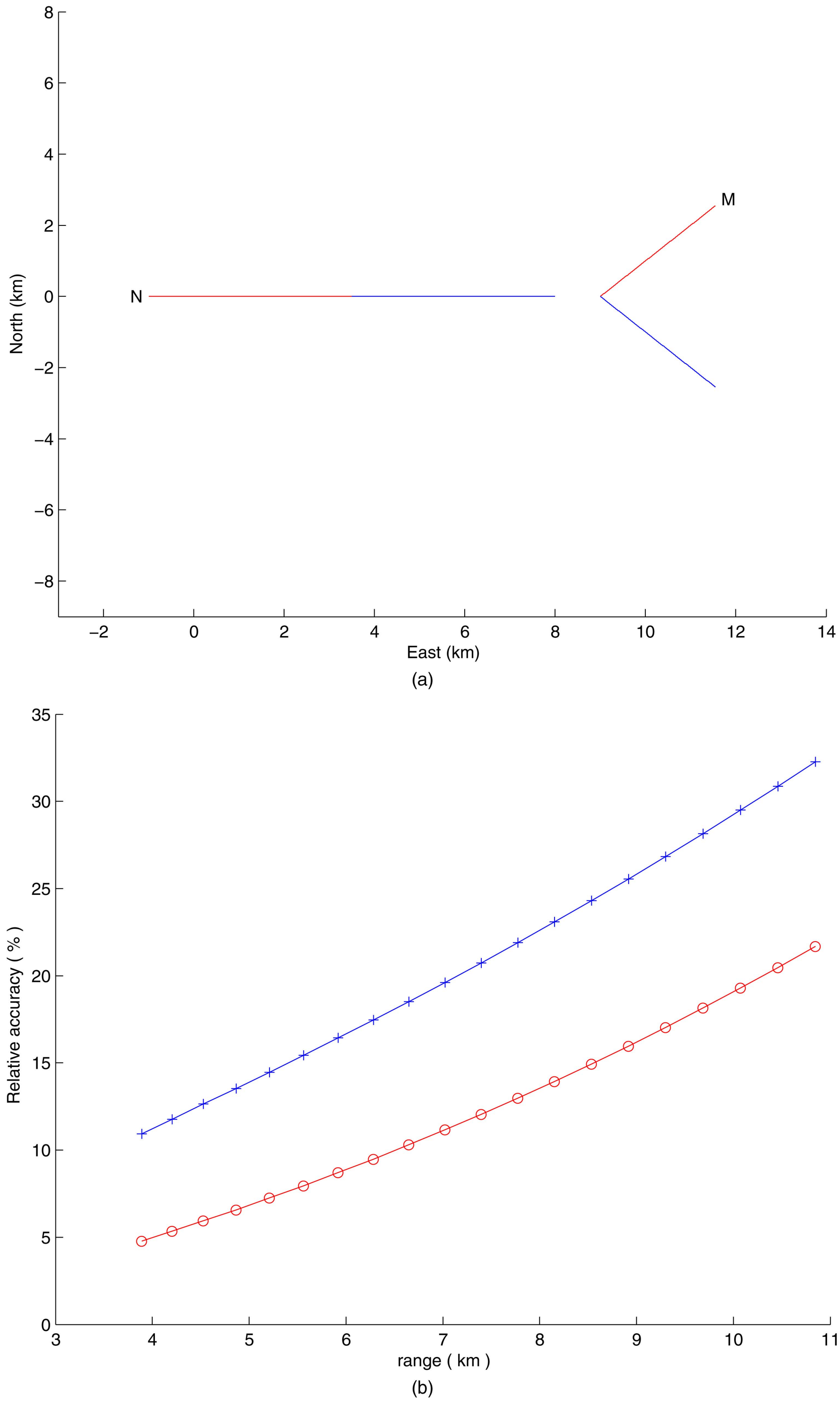


Fig. 10. (a) Typical escaping maneuvering platform trajectory for the third example. (b) The relative accuracies of the estimated ranges for the third example ("o" for the conventional BOTMA, "+" for the BOMTMA).

We deduce that

$$\nabla_{\tilde{X}}\theta_k(\tilde{X}) = \frac{1}{r^2(t_k)}\begin{bmatrix} -\Delta_x(t_k) \\ \Delta_t(t_k)\Delta_y(t_k) \\ -\Delta_t(t_k)\Delta_x(t_k) \end{bmatrix}.$$

The computation of the confidence bands is based on the Jacobians of the two following mappings

$$\tilde{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \mapsto Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \mapsto V(\rho) = \begin{bmatrix} v(\rho) \\ c(\rho) \end{bmatrix},$$

with

$$Y = \begin{bmatrix} \operatorname{atan}\left[\dfrac{\bar{\rho}\sin\beta_{t^*}}{x_2 - y_O(t^*)}\right] \\ \dfrac{\sqrt{(x_3 - \dot{x}_O)^2 + (x_4 - \dot{y}_O)^2}}{\sqrt{(\bar{\rho}\sin\beta_{t^*})^2 + [x_2 - y_O(t^*)]^2}} \\ \operatorname{atan}\left[\dfrac{x_3 - \dot{x}_O}{x_4 - \dot{y}_O}\right] \end{bmatrix} \quad \text{and}$$

$$V(\rho) = \begin{bmatrix} v(\rho) = \sqrt{(\rho y_2 \sin y_3 + \dot{x}_O)^2 + (\rho y_2 \cos y_3 + \dot{y}_O)^2} \\ c(\rho) = \operatorname{atan}\left[\dfrac{\rho y_2 \sin y_3 + \dot{x}_O}{\rho y_2 \cos y_3 + \dot{y}_O}\right] \end{bmatrix}.$$

The CRLB of $V(\rho)$ is then

$$B(V(\rho)) = J_2 J_1 B(\tilde{X}) J_1^{\mathrm{T}} J_2^{\mathrm{T}}$$

where J_1 is the Jacobian of the mapping $\tilde{X} \mapsto Y$ and J_2 is the Jacobian of the mapping $Y \mapsto V(\rho)$. We evaluate the confidence bands of each component of $v(\rho)$ and of $c(\rho)$ from the diagonal terms of $B(V(\rho))$.

ACKNOWLEDGMENT

We are grateful to Caroline Turner for having upgraded the writing of this paper.

REFERENCES

[1] Y. Bar-Shalom and X. R. Li
Estimation and Tracking: Principles, Techniques, and Software.
Artech House (1993).

[2] P. Blanc-Benon and J. M. Passerieux
Fusion of reduced-rank TMA estimates.
IEEE Transactions on Aerospace and Electronic Systems, **AES-36**, 3 (July 2000), 750–759.

[3] J. E. Dennis
A user's guide to nonlinear optimization algorithms.
In *Proceedings of IEEE*, **72**, 12 (Dec. 1984).

[4] K. C. Ho and Y. T. Chan
An asymptotically unbiased estimator for bearings-only and Doppler-bearing target motion analysis.
IEEE Transactions on Signal Processing, **54**, 3 (Mar. 2006), 809–821.

[5] J. Holst and A. Holtsberg
On target manoeuvres in bearings-only tracking.
In *Proceedings of the 25th Asilomar Conference on Circuits Systems*, (Nov. 1991).

[6] C. Jauffret and D. Pillon
Observability in passive target motion analysis.
IEEE Transactions on Aerospace and Electronic Systems, **AES-32**, 4 (Oct. 1996), 1290–1300.

[7] C. Jauffret, D. Pillon, and A. C. Pignol
Bearings-only TMA without observer maneuver.
In *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, (July 2008), 501–508.

[8] C. Jauffret, D. Pillon, and A. C. Pignol
Bearings-only maneuvering target motion analysis from a non maneuvering platform.
IEEE Transactions on Aerospace and Electronic Systems, **AES-46**, 4 (Oct. 2010), 1934–1949.

[9] C. Jauffret, D. Pillon, and A. C. Pignol
Leg-by-leg bearings-only TMA without observer maneuver.
In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, (July 2009), 2036–2043.

[10] J. P. Le Cadre and C. Jauffret
Discrete-time observability and estimability analysis for bearings-only target motion analysis.
IEEE Transactions on Aerospace and Electronic Systems, **AES-33**, 1 (Jan. 1997), 178–201.

[11] J. P. Le Cadre and S. Laurent-Michel
Optimizing the receiver maneuvers for bearings-only tracking.
Automatica, **35** (1999), 591–606.

[12] S. C. Nardone and V. J. Aidala
Observability criteria for bearings-only target motion analysis.
IEEE Transactions on Aerospace and Electronic Systems, **AES-17**, 2 (Mar. 1981), 162–166.

[13] S. C. Nardone, A. G. Lindgren, and K. F. Gong
Fundamental properties and performance of conventional bearings-only target motion analysis.
IEEE Transactions on Automatic Control, **AC-29**, 9 (Sept. 1984), 775–787.

[14] S. C. Nardone and M. L. Graham
A closed-form solution to bearings-only target motion analysis.
IEEE Journal of Oceanic Engineering, **OE-22**, 1 (Jan. 1997), 168–178.

[15] J. M. Passerieux and D. Van Cappel
Optimal observer maneuver for bearings-only tracking.
IEEE Transactions on Aerospace and Electronic Systems, **AES-34**, 3 (July 1998), 777–788.

[16] D. T. Pham
Some quick and efficient methods for bearings-only target motion analysis.
IEEE Transactions on Signal Processing, **SP-41**, 9 (Sept. 1993), 2737–2751.

[17] J. Ru, V. P. Jilkov, X. R. Li, and A. Bashi
Detection of target maneuver onset.
IEEE Transactions on Aerospace and Electronic Systems, **AES-45**, 2 (Apr. 2009), 536–554.

Claude Jauffret born in France on March 29, 1957, received the diplôme d'Etudes Appronfondies in applied mathematics from Saint Charles University, Marseille, France, in 1981, the Diplôme d'Ingénieur from Ecole Nationale Supérieure d'Informatique et de Mathématiques Appliqués de Grenoble, Grenoble, France, in 1983, the title of Docteur de l'Université in 1993 and the Habilitation à Diriger des Recherches from the Université de Toulon et du Var, France.

From November 1983 to November 1988, he worked on passive sonar systems, more precisely on target motion analysis at GERDSM, France. After a sabbatical year at the University of Connecticut (from November 1988 to December 1989) during which he worked of tracking problems in cluttered environment, he developed researches in tracking, data fusion, and extraction in CERDSM. Since September 1996, he has been at the Université de Toulon et du Var where he teaches statistical signal processing. His current researches are about observability, estimation in non linear systems as they appear in tracking problems.

He is an associate editor for *IEEE Transactions on Aerospace and Electronic Systems.*

Denis Pillon comes from a small French mountain called "Le Jura." He received the Diplôme d'Ingénieur from Ecole Supérieure de Physique et de Chimie Industrielle (ESPCI-Paris) in 1977 and the Diplôme d'Etudes Approfondies in probability theory from the Jussieu University, Paris in 1977.

From 1978 to 1979 he worked in the area of operational research at the Centre InterArmées de Recherche Opérationnelle (CIRO) agency of the ministère de la Défense. Since 1980 he has been employed by Thomson Compagny (now Thales Compagny), France. From 1980 to 1981 he worked on sonar array processing. From 1982 to 1986, he was consultant at the GERDSM for which he worked on sea trial analysis and TMA. From 1987 to 1989, he worked on tracking and track clustering and in 1990 he was appointed group leader of signal and data processing laboratory at Thomson Sintra company, Sophia Antipolis, France. From 1992 to 1999, he was consultant at the DCN sonar laboratory. He worked on new sonar arrays and various submarine combat systems. Since 2000, he has been sonar expert at Thales Underwater Systems (Sophia-Antipolis). His current researches are about passive sonar performances estimation, submarine and torpedo array processing, and long term prospective studies for submarine warfare.

Annie-Claude Pignol born in France on November 10, 1965, received the Diplôme d'Etudes Appronfondies in optics and image processing from the Université de Toulon et du Var, Toulon, France, in 1988 and the title of Docteur de l'Université in 1991 from the Université de Toulon et du Var, France.

Since September 1994, she has been at the Université de Toulon et du Var where she teaches electronic systems. Her researches were focused signal processing applied to biomedical systems before turning them to Target Motion Analysis.

Anomaly Detection using Context-Aided Target Tracking

JEMIN GEORGE

JOHN L. CRASSIDIS

TARUNRAJ SINGH

ADAM M. FOSBURY

The main objective of this work is to model and exploit available contextual information to provide a hypothesis on suspicious vehicle maneuvers. This paper presents an innovative anomaly detection scheme, which utilizes L1 tracking to perform L2/L3 data fusion, i.e., situation/threat refinement and assessment. The proposed concept involves a context-aided tracker called the *Con-Tracker*, a multiple-model adaptive estimator, and an L2/L3 hypothesis generator. The purpose of the Con-Tracker is to incorporate the contextual information into a traditional Kalman filter-based tracker in such a way that it provides a repeller or attractor characteristic to a specific region of interest. Any behavior of the vehicle that is inconsistent with the repeller or attractor characteristic of the current vehicle location would be classified as suspicious. Such inconsistent vehicle behavior would be directly indicated by a high measurement residual, which then may be used to estimate the process noise covariance associated with the context-aware model using a multiple-model adaptive estimator. Based on the rate of change of the estimated process noise covariance values, an L2/L3 hypothesis generator red-flags the target vehicle. Simulation results indicate that the proposed concept involving context-aided tracking enhances the reliability of anomaly detection.

Manuscript received October 9, 2009; revised May 26, 2010, November 13, 2010, and February 1, 2011, released for publication February 14, 2011.

Refereeing of this contribution was handled Peter Willett.

Authors' addresses: J. George, U.S. Army Research Laboratory, Networked Sensing & Fusion Branch, Adelphi, MD 20783; J. Crassidis and T. Singh, Department of Mechanical & Aerospace Engineering, University at Buffalo, State University of New York, Amherst, NY 14260; A. Fosbury, Johns Hopkins University, Applied Physics Laboratory, Laurel, MD 20723.

1. INTRODUCTION

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected normal behavior. Anomaly detection is extensively used in a wide variety of applications such as monitoring business news, epidemic or bioterrorism detection, intrusion detection, hardware fault detection, network alarm monitoring, and fraud detection [13]. Anomaly detection in target tracking is an essential tool in separating benign targets from intruders that pose a threat. This paper presents a new, innovative anomaly detection scheme using context-aided target tracking.

Various data, feature, and knowledge fusion strategies and architectures have been developed over the last several years for improving the accuracy, robustness, and overall effectiveness of anomaly detection technologies. Singh et al. [41] illustrate the capabilities of hidden Markov models (HMMs), combined with feature-aided tracking, for the detection of asymmetric threats. In [41], HMMs are integrated into feature-aided tracking using a transaction-based probabilistic model and a procedure analogous to Page's test is used for the quickest detection of abnormal events. An information fusion-based decision support tool is presented in [8] to aid the identification of a target carrying out a pattern of activity, which could be comprised of a wide variety of possible sub-activities. Barker et al. [8] propose the time series anomaly detection methods to process multi-modal sensor data, which are then integrated by a Bayesian information fusion algorithm to provide a probability that each candidate under observation is carrying out the target activity. While the traditional anomaly-based intrusion detection approach builds one global profile for normal activities and detects intrusions by comparing current activities with the normal profile, Salem and Karim [39] propose a context-based profiling methods for building more realistic normal profiles than global ones. Moreover, contextual information is also exploited to build attack profiles that can be used for diagnosis purposes. Jackson et al. [21] propose a cognitive fusion approach for detecting anomalies appearing in the behavior of dynamic self-organizing systems such as sensor networks, mobile ad hoc networks, and tactical battle management. Fusion of relevant sensor data, maintenance database information, and outputs from various diagnostic and prognostic technologies have proven effective in reducing false alarm rates, increasing confidence levels in early fault detection, and predicting time to failure or degraded condition requiring maintenance action. Roemer et al. [38] provide an overview of various aspects of data, information, and knowledge fusion, including the places where fusion should exist within a health management system, the different types of fusion architectures, and a number of different fusion techniques. Compared to these existing context-aided anomaly detection schemes, the proposed approach has five main advantages:

- Existing context-aided anomaly detection schemes are strictly observation-based while the proposed approach utilizes a dynamic model of the target. In current approaches, observations are compared to a nominal/begin target activity, while the proposed approach compares the target model to that of a nominal model.
- The presented approach can be easily modified so that the target model refinement is a byproduct of the proposed anomaly detection scheme.
- The dynamic target model can be used to predict future target states or activities.
- The proposed scheme is easily compatible with existing target tracking algorithms.
- The context-aided anomaly detection technique presented here is more general compared to existing methods that are tailored to a specific scenario.

While early tracking algorithms have relied almost exclusively on target location measurements provided by sensors such as radars [31], [40], more advanced techniques have incorporated information pertaining to the orientation, velocity, and acceleration of the target [18], [25]–[27], [43], [46]. This progression suggests that increasing the amount of information incorporated into the algorithm can improve the quality of the tracking process. In ground-based target tracking, a map of terrain features affecting target motion is usually available. A terrain-based tracking approach that accounts for the effects of terrain on target speed and direction of movement is presented in [36]. In [34], it has been shown that the incorporation of local contextual information, such as the terrain data, can significantly improve tracker performance. In recent years, researchers have explored the overt use of contextual information for improving state estimation in ground target tracking by incorporating them into the tracking algorithm as potential fields to provide a repeller or attractor characteristic to a specific region of interest [44], [45]. In [19], the local contextual information, termed "trafficability," incorporates local terrain slope, ground vegetation, and other factors to put constraints on the vehicle's maximum velocity. Simulation results given in [19] show that the use of trafficability can improve estimate accuracy in locations where the vehicle path is influenced by terrain features.

There exist several constrained target tracking algorithms. The kinematic constraints on target state provides information that can be processed as a pseudo-measurement to improve tracking performance. For example, Alouani [3] shows that the filter utilizing the kinematic constraint as a pseudo-measurement is unbiased when the system with the kinematic constraint is observable and the use of the kinematic constraint can increase the degree of observability of the system. Alouani and Blair [1], [2] propose a new formulation

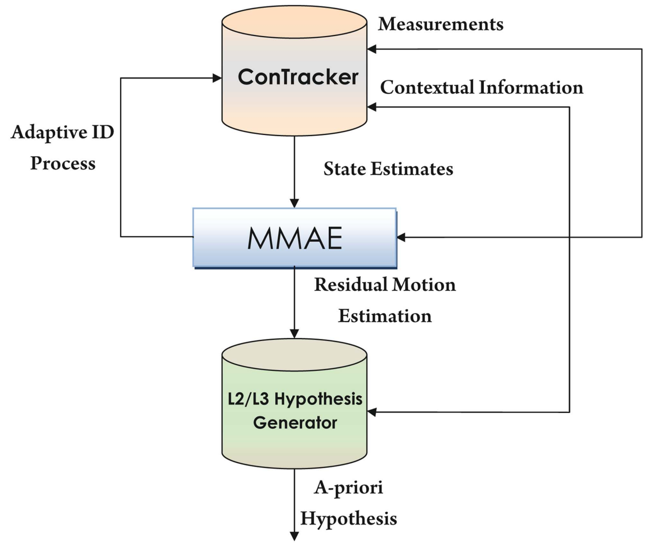


Fig. 1. System flowchart.

of the kinematic constraint for constant speed targets, which is shown to be unbiased and, under mild restriction, uniformly asymptotically stable. Though the proposed approach exploits contextual information to place constraints on target velocity, an explicit expression for the kinematic constraints on target state cannot be easily obtained since the contextual information depends on the current target position. Also, the use of a kinematic constraint as a pseudo-measurement would severely degrade the performance of the proposed anomaly detection scheme.

The main goal of this work is to exploit available contextual information to provide a hypothesis on suspicious vehicle maneuvers and perform L2/L3 data fusion,[1] i.e., situation and threat, refinement and assessment (see [24] for the Joint Directors of Laboratories' description of the various data fusion levels). Although the approach presented herein can be applied to any vehicle system, such as air-, ground- or sea-based vehicles, the particular application here involves maritime tracking and contextual information. For example, it is desired to "red-flag" a boat that approaches a restricted high-value unit area. Also, a vessel that is erratically zigzagging across a marked shipping channel may be red-flagged for suspicious activity. The process to provide a hypothesis of this notion is depicted in Fig. 1. The proposed concept involves exploiting the mathematically rigorous approaches of L1 tracking in an L2/L3 situation and threat, refinement and assessment scheme. In [37], a statistical anomaly detection scheme for maritime vessels using adaptive kernel density estimation scheme is presented. The methodology pre-

[1]Level 1 (L1) fusion is aimed at combining sensor data to obtain accurate system states, Level 2 (L2) fusion dynamically attempts to develop a description of relationships among entities and events, and Level 3 (L3) fusion projects the current situation into the future to draw inferences about threats.

sented here consist of three main components: a context-aided tracker, called "Con-Tracker," a Multiple-Model Adaptive Estimator, and a hypothesis generator.

The Con-Tracker combines contextual information with L1 measurement information to provide state estimates (position and velocity). Depth, marked shipping channel locations, and high-value unit information are a few examples of contextual information pertaining to the particular maritime scenario considered here. The purpose of the Con-Tracker is to use the contextual information in such a way that it provides a repeller or attractor characteristic to each region developed through a grid-spaced map of a particular area of interest. In the propagation stage of the Con-Tracker, vehicle states are propagated according to the repeller or attractor characteristic of the current location of the vehicle. Any behavior of the vehicle that is inconsistent with the repeller or attractor characteristic of the current location would be classified as suspicious. Such inconsistent vehicle behavior would be directly indicated by a high measurement residual, which may then be used to estimate the process noise covariance associated with the target model. Thus, Con-Tracker accuracy is not only a function of the contextual information provided; its performance also depends on the usual Kalman "tuning" issue, i.e., determination of the process noise covariance [4], [15]. The tuning process is a function of the actual vehicle motion, which can vary. This variation is the key to the hypothesis generator. This is best explained by an example. Suppose that when a vehicle is heading towards a high-value unit, the contextual information incorporated into the Con-Tracker would repel the vehicle away from the high-value unit during the propagation stage of the tracker. However, if the vehicle still proceeds towards the high-value unit, which is shown directly through the measurements of the vehicle location, then in order to provide good tracker characteristics, a large value of process noise covariance must be chosen, i.e., tuned.

The aforementioned tuning issue is usually performed in an ad-hoc manner. However, mathematical tools can be used to automatically tune the tracker. Multiple-model estimation schemes are useful for the process noise identification (tuning) problem. Multiple-model estimation approaches run parallel trackers, where each tracker uses a different value for the process noise covariance. The covariance is identified using the likelihood function of the measurement residuals, which provides weights on each individual tracker [4]. There exist several multiple-model-based target tracking schemes, such as the Multiple-Model Adaptive Estimator (MMAE), Interacting Multiple Model (IMM), Adaptive-Interacting Multiple Model (A-IMM), and Variable Structure-Interacting Multiple Model (VS-IMM). All of these approaches are based on a near-constant velocity model in some form. Kastella and Kreucher [23] describe the design and implementation of a multiple-model nonlinear filter (MMNLF) for ground target tracking using ground moving target indicator (GMTI) radar measurements. While target tracking in an arbitrarily dense multitarget-multisensor environment is a formidable problem, the interacting multiple model algorithm techniques have been shown to achieve reliable tracking performance [6], [10], [16], [28]–[30]. The IMM estimator, originally proposed by Blom [9], is a suboptimal hybrid filter that was shown to achieve an excellent compromise between performance and complexity. Munir and Atherton [17], [32], [33] describe an A-IMM algorithm for maneuvering target tracking. The algorithm proposed in [33] estimates the target acceleration using a two-stage Kalman estimator, and the estimated acceleration value is fed to the subfilters in an IMM algorithm, where the subfilters have different acceleration parameters. A detailed survey of existing IMM methods for target tracking problems is presented in [30].

The main difference between IMM-based approaches and MMAE schemes is that IMM involves interaction between the models that require the explicit knowledge of transition probabilities between the modes. Since the calculation of transition probabilities could be computationally expensive, an MMAE approach is utilized here for the selection of appropriate process noise covariance. The MMAE scheme implemented here consists of a bank of Con-Trackers, each with a different process noise covariance. Assuming the estimated process noise covariance values are consistent with the truth, a small value of process noise covariance corresponds to a case where the context-aware target model is an accurate representation of the true target, and a large value of process noise covariance indicates that the context-aware target model is a poor representation of the truth and the target does not comply with the available contextual information. The process noise covariance is estimated as a weighted sum of all the process noise covariances used and the weight associated with each covariance is calculated using the likelihood of the process noise covariances conditioned on the current-time measurement-minus-estimate residual. The estimated covariance is incorporated into an L2/L3 hypothesis scheme that provides a hypothesis on whether or not a vehicle motion should be alerted to an analyst. The L2/L3 hypothesis generator red-flags the vehicle based on the rate of change of the process noise covariance and the contextual information provided. Details of these processes are provided in the subsequent sections.

2. CON-TRACKER

The main difference between a traditional tracker and the context-aided Con-Tracker is that the target model used in the Con-Tracker accounts for the local contextual information. The local contextual information is incorporated into the Con-Tracker model as trafficability values. Trafficability is a value between zero and one, where zero indicates a region that is

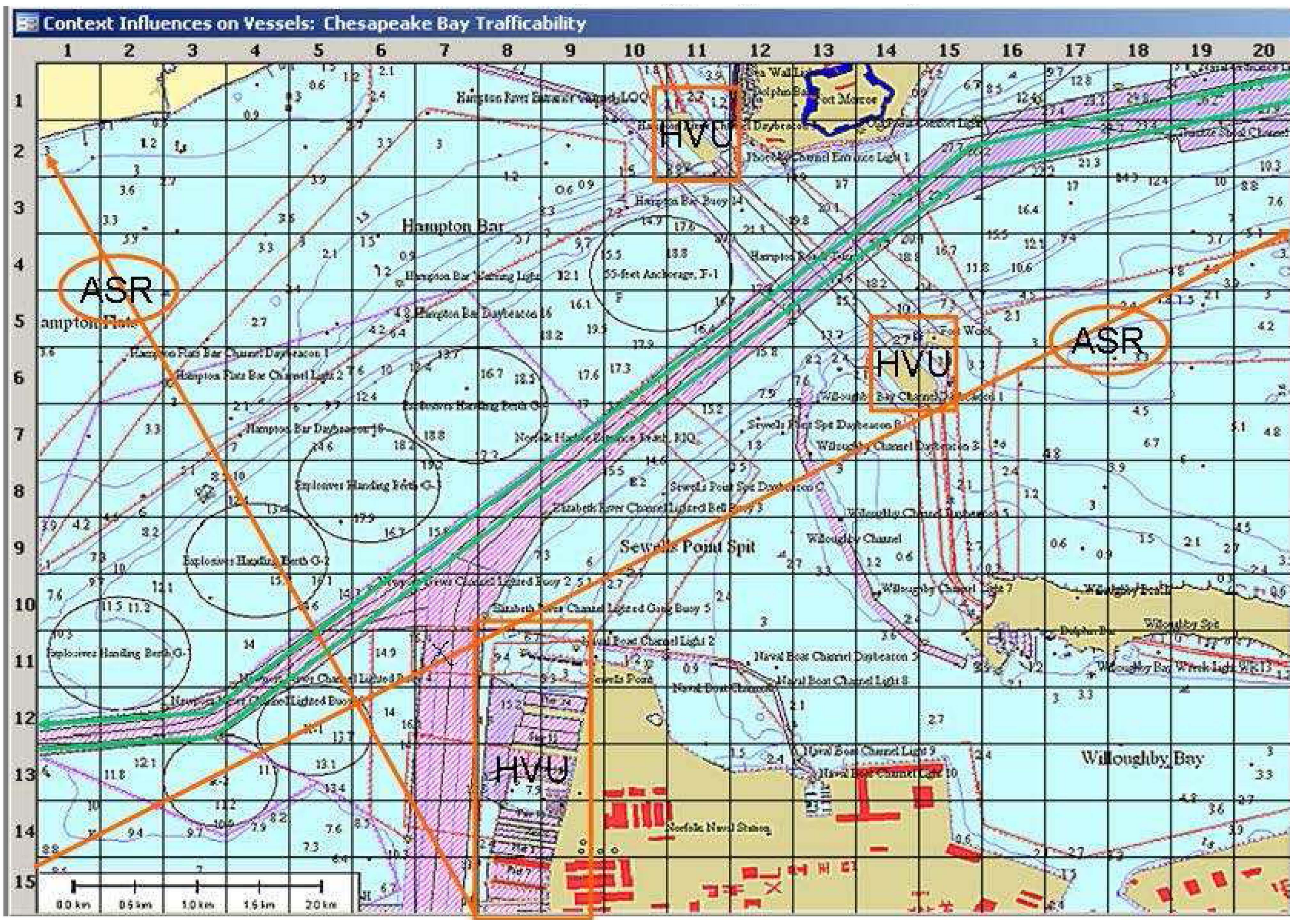


Fig. 2. Maritime trafficability values database.

not traversable and one indicates a region that is completely traversable. For the maritime applications considered here, these trafficability values are based on local traversability information and accounts for the following four "contextual" data:

- Depth information,
- marked channel information,
- anti-shipping reports (ASR), and
- locations of high-value units (HVU).

The individual trafficability values corresponding to each contextual information are combined into a single value, which is used to indicate the repeller or attractor characteristic of a specific region. Details of this procedure are given next.

First, a particular area of interest is divided into a grid-field, similar to a 15×20 grid-field, as shown in Fig. 2. In Fig. 2, the purple channels indicate marked shipping lanes. As shown in Fig. 2, the area of interest contains three high-value units centered around cells $(2,11)$, $(6,14)$, and $(11\ldots15,8)$. The area also contains two anti-shipping areas centered about cells $(4,2)$ and $(5,17)$. Finally, low-depth areas are mainly indicated using different shades of brown. According to the vehicle type that is being tracked, a single trafficability value, ν_i, is assigned to each cell. This variable is a decimal value between 0 and 1 and corresponds to the fraction of maximum velocity that the vehicle can attain in that grid location. For example, the grid cell $(10,17)$ has a trafficability of zero due to the depth information, and therefore, the vessels are supposed to avoid and navigate around this particular cell.

Trafficability data is also used to deflect the direction of target motion given by the past velocity information. In order to implement this, at each propagation stage in the Con-Tracker, we consider a 3×3 trafficability grid-field that depends on the current vehicle position. For example, if the vehicle is located in cell $(13,3)$, the 3×3 trafficability grid-field consists of cells $(12,2)$, $(12,3)$, $(12,4)$, $(13,2)$, $(13,3)$, $(13,4)$, $(14,2)$, $(14,3)$, and $(14,4)$. A generic representation of the 3×3 trafficability grid-field is shown in Fig. 3. The vehicle is assumed to be located in square 5 of the 3×3 grid. The 3×3 grid is continually re-centered about the vehicle as it moves throughout the region so that it is always located in the center (square 5) of the 3×3 trafficability grid-field. In Fig. 3, the unit vector $\hat{\mathbf{G}}_{tg} \in \mathbb{R}^2$ represents the preferred direction of the vehicle strictly based on the trafficability information of the surrounding cells, $\hat{\mathbf{G}}^- \in \mathbb{R}^2$ is a unit vector in the direction of target motion given by the past state information, and the unit vector $\hat{\mathbf{G}}^+ \in \mathbb{R}^2$ represents the nudged velocity direction. A

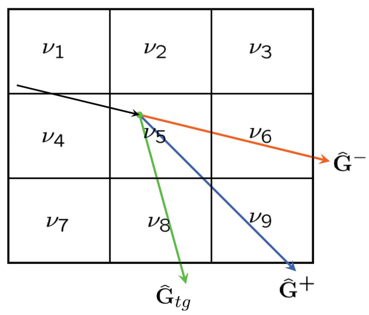


Fig. 3. 3×3 Trafficability grid-field: $\hat{\mathbf{G}}^-$ is the direction of target velocity, $\hat{\mathbf{G}}_{tg}$ is the preferred direction of target, $\hat{\mathbf{G}}^+$ is the nudged velocity direction, and ν_i indicate the trafficability of ith cell.

preferred direction based on the velocity constraint is calculated as

$$\hat{\mathbf{G}}_{tg} = \frac{\sum_j(\nu_j\hat{\mathbf{G}}_j)}{\left\|\sum_j(\nu_j\hat{\mathbf{G}}_j)\right\|} \tag{1}$$

where $j\in\{1,2,3,4,6,7,8,9\}$. The unit vector $\hat{\mathbf{G}}_j\in\mathbb{R}^2$ points from the current vehicle location to the center of square j. Now the nudged velocity direction of motion is given as

$$\hat{\mathbf{G}}^+ = \frac{\hat{\mathbf{G}}^- + \mu\hat{\mathbf{G}}_{tg}}{\|\hat{\mathbf{G}}^- + \mu\hat{\mathbf{G}}_{tg}\|} \tag{2}$$

where μ is a weighting coefficient that is calculated based on the absolute average difference in the trafficability values between the current location and the surrounding feasible locations

$$\mu = \frac{\sum_j|\nu_j-\nu_5|}{8}. \tag{3}$$

Note that the proposed technique for determining the nudged velocity direction is chosen because it is least expensive in terms of computational requirements.

2.1. Filter Algorithm

The theoretical developments of the Con-Tracker algorithm, which is based on the standard near-constant velocity tracker, are now shown. The state vector used in the filter is $\mathbf{x}\in\mathbb{R}^4$, i.e.,

$$\mathbf{x} = [\lambda \quad \phi \quad v_\lambda \quad v_\phi]^T \tag{4}$$

where λ, ϕ, v_λ, and v_ϕ are the longitude and latitude locations of the target vehicle and the corresponding rates. In the case of the near-constant velocity models used in the α-β tracker, zero-mean Gaussian white process noise is added to the model to account for the possible variations in velocity [7], [22]. Our approach modifies this concept by using the following discrete-time model

$$\mathbf{x}_{k+1} = \left.\begin{bmatrix} \lambda + v_\lambda\Delta t \\ \phi + v_\phi\Delta t \\ \nu_5\sqrt{v_\lambda^2+v_\phi^2}\cos\theta \\ \nu_5\sqrt{v_\lambda^2+v_\phi^2}\sin\theta \end{bmatrix}\right|_k + \mathbf{w}_k \tag{5}$$

where

$$E[\mathbf{w}_k\mathbf{w}_k^T] = \Upsilon Q_k \Upsilon^T$$
$$= \begin{bmatrix} \frac{\Delta t^3}{3}q_{1_k} & 0 & \frac{\Delta t^2}{2}q_{1_k} & 0 \\ 0 & \frac{\Delta t^3}{3}q_{2_k} & 0 & \frac{\Delta t^2}{2}q_{2_k} \\ \frac{\Delta t^2}{2}q_{1_k} & 0 & \Delta t q_{1_k} & 0 \\ 0 & \frac{\Delta t^2}{2}q_{2_k} & 0 & \Delta t q_{2_k} \end{bmatrix}$$

with

$$\Upsilon\in\mathbb{R}^{4\times 2} \qquad \text{and} \qquad Q_k \equiv \begin{bmatrix} q_{1_k} & 0 \\ 0 & q_{2_k} \end{bmatrix}.$$

The angle θ, the angle between the velocity vector and the local y-axis (north axis), defines the assumed direction of motion of the vehicle, $\hat{\mathbf{G}}^+$, i.e.,

$$\hat{\mathbf{G}}^+ = [\cos\theta \quad \sin\theta]^T. \tag{6}$$

The unit vector $\hat{\mathbf{G}}^+$ is determined using the trafficability data as explained in (2). The coefficient ν_5 is the trafficability of the current cell. The $\sqrt{v_\lambda^2+v_\phi^2}$ term is simply the magnitude of the vehicle velocity and the trigonometric terms are used to project this value onto the appropriate axes. When no trafficability information is present, ν_5 defaults to one, and the trigonometric terms are given by

$$\cos\theta = \frac{v_\lambda}{\sqrt{v_\lambda^2+v_\phi^2}}, \qquad \sin\theta = \frac{v_\phi}{\sqrt{v_\lambda^2+v_\phi^2}} \tag{7}$$

which reduce (5) to the standard near-constant velocity model used in the α-β tracker. Notice that the $\hat{\mathbf{G}}^-$ in (2) is given as

$$\hat{\mathbf{G}}^- = \begin{bmatrix} \frac{v_\lambda}{\sqrt{v_\lambda^2+v_\phi^2}} & \frac{v_\phi}{\sqrt{v_\lambda^2+v_\phi^2}} \end{bmatrix}^T. \tag{8}$$

The measurement vector is assumed to be

$$\mathbf{y} = [\lambda \quad \phi]^T + [\mathrm{v}_\lambda \quad \mathrm{v}_\phi]^T \tag{9}$$

where $\mathbf{v} = [\mathrm{v}_\lambda \ \mathrm{v}_\phi]^T$ is the zero-mean Gaussian white-noise sequence, i.e., $E[\mathbf{v}] = \mathbf{0}$ and $E[\mathbf{v}_j\mathbf{v}_k^T] = R\delta_{jk}$. Let

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

then the measurement equation can be written as

$$\mathbf{y}_k = H\mathbf{x}_k + \mathbf{v}_k. \tag{10}$$

The near-constant velocity target model without the velocity nudging can be written in concise form as

$$\mathbf{x}_{k+1} = \Psi\mathbf{x}_k + \mathbf{w}_k \tag{11}$$

where

$$\Psi = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The estimation error covariance is defined as $P_k = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T]$, and the following equations are used to propagate and update the error covariance matrix

$$P_{k+1}^- = \Psi P_k^+ \Psi^T + \Upsilon Q_k \Upsilon^T \tag{12}$$

$$P_k^+ = [I - K_k H_k] P_k^- \tag{13}$$

where $P_k^- = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T]$ is the *a priori* error covariance and $P_k^+ = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)^T]$ is the *a posteriori* error covariance. The matrix K_k is the usual Kalman gain and can be calculated using

$$K_k = P_k^- H^T [H P_k^- H^T + R]^{-1}. \tag{14}$$

The vector $\hat{\mathbf{x}}_k^-$ is referred to as the *a priori* state estimate and the vector $\hat{\mathbf{x}}_k^+$ is referred to as the *a posteriori* state estimate. The estimates are propagated and updated using

$$\hat{\mathbf{x}}_{k+1}^- = \begin{bmatrix} \hat{\lambda}^+ + \hat{v}_\lambda^+ \Delta t \\ \hat{\phi}^+ + \hat{v}_\phi^+ \Delta t \\ \nu\sqrt{(\hat{v}_\lambda^+)^2 + (\hat{v}_\phi^+)^2}\cos\theta \\ \nu\sqrt{(\hat{v}_\lambda^+)^2 + (\hat{v}_\phi^+)^2}\sin\theta \end{bmatrix}\Bigg|_k \tag{15}$$

$$\mathbf{x}_k^+ = \mathbf{x}_k^- + K_k[\mathbf{y}_k - H\mathbf{x}_k^-]. \tag{16}$$

The Con-Tracker algorithm is summarized in Table I. Note that the Con-Tracker algorithm is very similar to that of a traditional Kalman filter-based tracking algorithm without the velocity nudging during the propagation stage. Since the process noise is added to the context-aware near-constant velocity model to account for variations in velocity, the process noise covariance Q_k indicates the accuracy of target model in (5), i.e., how well a target complies with the given contextual information and the constant velocity assumption. If the target vehicle follows the model precisely, then Q_k would be fairly small. If the vehicle maneuvers are erratic and inconsistent with the model, then the process noise covariance would be large. Since one does not know the precise value of the process noise covariance, an MMAE approach is implemented to estimate the process noise covariance based on the measurement residual.

TABLE I
Summary of Con-Tracker Algorithm

Initialize	$\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0^-,\ P_0^- = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0^-)(\mathbf{x}_0 - \hat{\mathbf{x}}_0^-)^T]$
Kalman Gain	$K_k = P_k^- H^T [H P_k^- H^T + R]^{-1}$
Update	$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k[\mathbf{y}_k - H\hat{\mathbf{x}}_k^-]$
	$P_k^+ = [I - K_k H_k] P_k^-$
Velocity Nudging	$\hat{\mathbf{G}}^- = \left[\dfrac{\hat{v}_\lambda^+}{\sqrt{(\hat{v}_\lambda^+)^2 + (\hat{v}_\phi^+)^2}} \quad \dfrac{\hat{v}_\phi^+}{\sqrt{(\hat{v}_\lambda^+)^2 + (\hat{v}_\phi^+)^2}}\right]\Bigg|_k$ $\hat{\mathbf{G}}_{tg} = \dfrac{\sum_j(\nu_j \hat{\mathbf{G}}_j)}{\left\|\sum_j(\nu_j \hat{\mathbf{G}}_j)\right\|}$ $\hat{\mathbf{G}}^+ = \hat{\mathbf{G}}^- + \mu\hat{\mathbf{G}}_{tg}$ $[\cos\theta \quad \sin\theta]^T = \hat{\mathbf{G}}^+$
Propagation	$P_{k+1}^- = \Psi P_k^+ \Psi^T + \Upsilon Q_k \Upsilon^T$ $\hat{\mathbf{x}}_{k+1}^- = \begin{bmatrix} \hat{\lambda}^+ + \hat{v}_\lambda^+ \Delta t \\ \hat{\phi}^+ + \hat{v}_\phi^+ \Delta t \\ \nu\sqrt{(\hat{v}_\lambda^+)^2 + (\hat{v}_\phi^+)^2}\cos\theta \\ \nu\sqrt{(\hat{v}_\lambda^+)^2 + (\hat{v}_\phi^+)^2}\sin\theta \end{bmatrix}\Bigg|_k$

3. MULTIPLE-MODEL ADAPTIVE ESTIMATION

A brief overview of the MMAE approach is presented in this section. More details on the formulation of MMAE can be found in [4], [11], [42]. MMAE is a recursive estimator that uses a bank of filters that depend on some unknown parameters. In the problem under consideration, these unknown parameters are the process noise variances (diagonal elements of the process noise covariance) denoted by the vector $\mathbf{q}_k = [q_{1_k} \ \ q_{2_k}]^T$. For notational simplicity, the subscript k is omitted for $\mathbf{q}$. Initially, a set of distributed elements is generated from some known probability density function (pdf) of $\mathbf{q}$, denoted by $p(\mathbf{q})$, to give $\{\mathbf{q}^{(\ell)};\ \ell = 1,\ldots,M\}$. Here, M denotes the number of filters in the filter bank. The goal of the estimation process is to determine the conditional pdf of the ℓth element $\mathbf{q}^{(\ell)}$ given the current-time measurement $\mathbf{y}_k$. Application of Bayes' law yields

$$p(\mathbf{q}^{(\ell)} \mid \mathbf{Y}_k) = \frac{p(\mathbf{Y}_k, \mathbf{q}^{(\ell)})}{p(\mathbf{Y}_k)} = \frac{p(\mathbf{Y}_k \mid \mathbf{q}^{(\ell)})p(\mathbf{q}^{(\ell)})}{\sum_{j=1}^{M} p(\mathbf{Y}_k \mid \mathbf{q}^{(j)})p(\mathbf{q}^{(j)})} \tag{17}$$

where $\mathbf{Y}_k$ denotes the sequence $\{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_k\}$. The probabilities $p(\mathbf{q}^{(\ell)} \mid \mathbf{Y}_k)$ can be written as

$$p(\mathbf{q}^{(\ell)} \mid \mathbf{Y}_k) = \frac{p(\mathbf{y}_k, \mathbf{Y}_{k-1}, \mathbf{q}^{(\ell)})}{p(\mathbf{y}_k, \mathbf{Y}_{k-1})} = \frac{p(\mathbf{y}_k \mid \mathbf{Y}_{k-1}, \mathbf{q}^{(\ell)})p(\mathbf{Y}_{k-1}, \mathbf{q}^{(\ell)})}{p(\mathbf{y}_k, \mathbf{Y}_{k-1})}.$$

Since $p(\mathbf{Y}_{k-1}, \mathbf{q}^{(\ell)}) = p(\mathbf{q}^{(\ell)} \mid \mathbf{Y}_{k-1})p(\mathbf{Y}_{k-1})$, $p(\mathbf{q}^{(\ell)} \mid \mathbf{Y}_k)$ can be written as

$$p(\mathbf{q}^{(\ell)} \mid \mathbf{Y}_k) = \frac{p(\mathbf{y}_k \mid \mathbf{Y}_{k-1}, \mathbf{q}^{(\ell)})p(\mathbf{q}^{(\ell)} \mid \mathbf{Y}_{k-1})p(\mathbf{Y}_{k-1})}{p(\mathbf{y}_k \mid \mathbf{Y}_{k-1})p(\mathbf{Y}_{k-1})}.$$

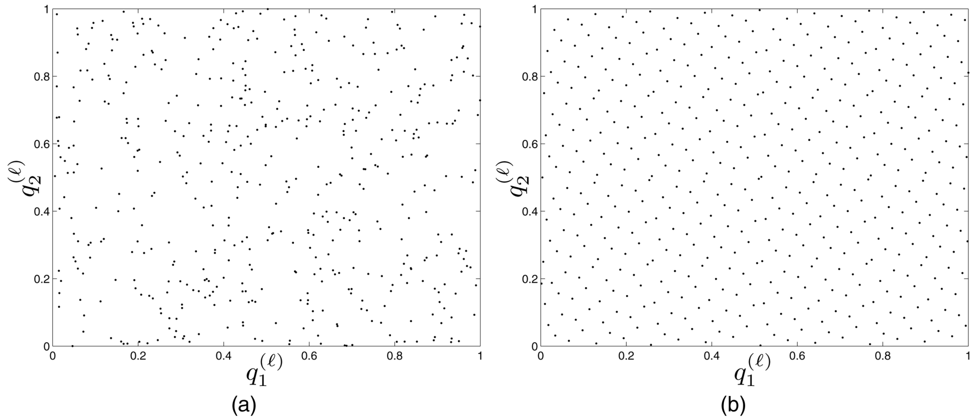


Fig. 4. Uniform distribution and Hammersley quasi-random sequence comparison. (a) Uniform distribution. (b) Hammersley quasi-random sequence.

Now the probabilities $p(\mathbf{q}^{(\ell)} \mid \mathbf{Y}_k)$ can be computed through [4]

$$p(\mathbf{q}^{(\ell)} \mid \mathbf{Y}_k) = \frac{p(\mathbf{y}_k \mid \hat{\mathbf{x}}_k^{-(\ell)})p(\mathbf{q}^{(\ell)} \mid \mathbf{Y}_{k-1})}{\sum_{j=1}^{M}[p(\mathbf{y}_k \mid \hat{\mathbf{x}}_k^{-(j)})p(\mathbf{q}^{(j)} \mid \mathbf{Y}_{k-1})]} \tag{18}$$

since $p(\mathbf{y}_k, \mid \mathbf{Y}_{k-1}, \mathbf{q}^{(\ell)})$ is given by $p(\mathbf{y}_k \mid \hat{\mathbf{x}}_k^{-(\ell)})$ in the Kalman recursion. The recursion formula can be cast into a set of defined weights $\varpi_k^{(\ell)}$, so that

$$\varpi_k^{(\ell)} = \varpi_{k-1}^{(\ell)} p(\mathbf{y}_k \mid \hat{\mathbf{x}}_k^{-(\ell)}) \tag{19}$$

$$\varpi_k^{(\ell)} \leftarrow \frac{\varpi_k^{(\ell)}}{\sum_{j=1}^{M} \varpi_k^{(j)}} \tag{20}$$

where $\varpi_k^{(\ell)} \equiv p(\mathbf{q}^{(\ell)} \mid \tilde{\mathbf{y}}_k)$. The weights at time t_0 are initialized to $\varpi_0^{(\ell)} = 1/M$ for $\ell = 1,2,\ldots,M$. The convergence properties of the MMAE are shown in [4], which assumes ergodicity in the proof. The ergodicity assumptions can be relaxed to asymptotic stationarity and other assumptions are even possible for non-stationary situations [5]. The conditional mean estimate is the weighted sum of the parallel filter estimates

$$\hat{\mathbf{x}}_k^- = \sum_{j=1}^{M} \varpi_k^{(j)} \hat{\mathbf{x}}_k^{-(j)}. \tag{21}$$

Also, the error covariance of the state estimate can be computed using

$$P_k^- = \sum_{j=1}^{M} \varpi_k^{(j)} [\{P_k^-\}^{(j)} + (\hat{\mathbf{x}}_k^{-(j)} - \hat{\mathbf{x}}_k^-)(\hat{\mathbf{x}}_k^{-(j)} - \hat{\mathbf{x}}_k^-)^T]. \tag{22}$$

The specific estimate for $\mathbf{q}$ at time t_k, denoted by $\hat{\mathbf{q}}_k$, and error covariance, denoted by $\mathcal{P}_k$, are given by

$$\hat{\mathbf{q}}_k = \sum_{j=1}^{M} \varpi_k^{(j)} \mathbf{q}^{(j)} \tag{23a}$$

$$\mathcal{P}_k = \sum_{j=1}^{M} \varpi_k^{(j)} (\mathbf{q}^{(j)} - \hat{\mathbf{q}}_k)(\mathbf{q}^{(j)} - \hat{\mathbf{q}}_k)^T. \tag{23b}$$

Equation (23b) can be used to define 3σ bounds on the estimate $\hat{\mathbf{q}}_k$.

At time t_0, all the filters have the same weight associated with them and there are many possibilities for the initial distribution of the process noise covariance parameters. A simple approach is to assume a uniform distribution. We instead choose a Hammersley quasi-random sequence [20] due to its well-distributed pattern. A comparison between the uniform distribution and the Hammersley quasi-random sequence for 500 elements is shown in Fig. 4. Clearly, the Hammersley quasi-random sequence provides a better "spread" of elements than the uniform distribution. In low dimensions, the multidimensional Hammersley sequence quickly "fills up" the space in a well-distributed pattern. However, for very high dimensions, the initial elements of the Hammersley sequence can be very poorly distributed. Only when the number of sequence elements is large enough relative to the spatial dimension, the sequence is properly behaved. This is not much of a concern for the process noise covariance adaption problem since the dimension of the elements will be much larger than the dimension of the unknown process noise parameters.

4. L2/L3 HYPOTHESIS GENERATOR

As mentioned in Section 2, the estimated process noise covariance is indicative of how well the target vehicle follows the context-aware near-constant velocity model. If the target vehicle follows the model precisely, then the estimated process noise covariance would be fairly small, and if the vehicle maneuvers are erratic and inconsistent with the model, then the process noise covariance would be large. The incorporation of contextual data into the model allows variations in target vehicle velocity that are consistent with the given contextual information. For example, if the target vehicle in cell (7,13) of Fig. 2 that is traveling toward cell (5,15) makes a sharp right turn to avoid the high value unit in cell (6,14), then the sudden change in the vehicle's velocity is consistent with the contextual data provided, and therefore, would not result in an increase in the estimated process noise covariance. However, if the target vehicle in cell $(6,3)$ that is traveling toward cell $(3,1)$ continues to travel in a straight line with a constant velocity, then there would be an increase in the estimated process noise covariance and the vehicle would be red-flagged despite its consistent behavior in accordance with the near-constant velocity model. This is because its passage into cell $(5,2)$ is in contrast to the anti-shipping activities reported in that area. Thus, a hypothesis on suspicious vehicle maneuvers can be synthesized based on the change in estimated process noise covariance.

The near-constant velocity model combined with the trafficability information is given by

$$\mathbf{x}_{k+1} = \left.\begin{bmatrix} \lambda + v_\lambda \Delta t \\ \phi + v_\phi \Delta t \\ \nu_5 \sqrt{v_\lambda^2 + v_\phi^2}\cos\theta \\ \nu_5 \sqrt{v_\lambda^2 + v_\phi^2}\sin\theta \end{bmatrix}\right|_k + \mathbf{w}_k. \tag{24}$$

Any abrupt maneuver of the target vehicle that is inconsistent with the context-aware model can be treated as system process noise. This would, in turn, result in a sudden increase in the process noise covariance estimated by the MMAE. The two main objectives of the L2/L3 hypothesis generator is to red-flag a vehicle based on the anomalies in its behavior that are indicated by the change in process noise covariance and identify the reason behind the red-flagging.

Since any anomaly in target behavior is indicated by a change in estimated process noise covariance, the proposed red-flagging algorithm is based on two sets of process noise covariance values. One set, $\{\hat{q}_{1_k}, \hat{q}_{2_k}\}$, is the MMAE estimate based on the Con-Tracker measurement residual values and the second set, $\{\breve{q}_{1_k}, \breve{q}_{2_k}\}$, is a second pair of MMAE estimates obtained using the standard Kalman filter-based tracker. The only difference between these two trackers is that the standard Kalman filter-based tracker does not make use of any contextual information. The second set of estimates, $\{\breve{q}_{1_k}, \breve{q}_{2_k}\}$, is used to normalize the first set of process noise covariance values. The normalized process noise covariances values are given as

$$\bar{q}_{1_k} = \frac{\hat{q}_{1_k}}{\breve{q}_{1_k}}, \qquad \bar{q}_{2_k} = \frac{\hat{q}_{2_k}}{\breve{q}_{2_k}}. \tag{25}$$

Normalization would eliminate any minor deviations in the process noise covariance values due to additive measurement noise. It also helps to clearly identify any abrupt maneuver of the target vehicle that is inconsistent with the given contextual information. After normalizing the elements of the process noise covariance matrix, their Euclidian norm is calculated as

$$\|q_k\| = \sqrt{(\bar{q}_{1_k})^2 + (\bar{q}_{2_k})^2}. \tag{26}$$

The rate of change of the normalized process noise covariance norm can be calculated as

$$\Delta q_k = \frac{1}{\Delta t}[\|q_k\| - \|q_{k-1}\|]. \tag{27}$$

The "change" in process noise covariance indicates the "occurrence" of target activity that is inconsistent with the prior knowledge. Therefore, a vehicle is red-flagged if the rate of change on the normalized process noise covariance norm is greater than a prescribed threshold, i.e.,

$$\Delta q_k > \Delta q_{\max} \Rightarrow \text{Red-Flag}. \tag{28}$$

Considering the rate of change of the normalized process noise covariance norm instead of the absolute magnitude helps to circumvent the slow transient response of the MMAE and thus, to avoid red-flagging a target long after the occurrence of an anomaly.

A second red-flagging algorithm can be formulated based on a simple χ^2-test [7], [12], [35]. Suppose that a measurement residual is defined by $\mathbf{e}_k = \mathbf{y}_k - H\mathbf{x}_k^-$, where $\mathbf{y}_k$ is the measurement and $H\mathbf{x}_k^-$ is its corresponding estimate. For our case, the length of the measurement vector is $m = 2$, corresponding to longitude and latitude coordinates. The theoretically correct covariance associated with $\mathbf{e}_k$, denoted by E_k, can be derived from the Kalman filter equations, i.e., it is known from the Kalman tracking process. Define the following normalized error square (NES)

$$\varepsilon_k = \mathbf{e}_k^T \mathbf{E}_k \mathbf{e}_k. \tag{29}$$

The NES can be shown to have a χ^2 distribution with m degrees of freedom. A suitable check for the NES is to numerically show that the following condition is met with some level of confidence

$$E\{\varepsilon_k\} = m. \tag{30}$$

Typically, one writes the χ^2 variable with its degrees of freedom as χ_2^2. A probability region can be constructed by cutting off the percent-difference upper tail. For example, a 99% probability region for a χ^2 variance can be taken as the one-sided probability region (cutting off the 1% upper tail)

$$[0, \quad \chi_2^2(0.99)] = [0, \quad 9.210]. \tag{31}$$

Other values can be found on page 84 of [7]. If the calculated χ^2 value from (29) falls within this region, then an χ^2 test indicates that the vehicle follows the Con-Tracker model with a high confidence of 99% and should not be red-flagged.

The red-flagging reasoner deals with identifying the contextual information that is conflicting with the current target vehicle location. For example, the grid cell $(2,11)$ of Fig. 2 has a trafficability of zero due to the high-value unit location. Therefore, if a vehicle is located in cell $(2,11)$, then the conflicting contextual information is the high-value unit locations. Since the

Con-Tracker is assumed to have access to all the contextual information, the simplest red-flagging reasoner can be synthesized by identifying which of the four contextual data contributes to the zero trafficability at the current location. The main assumption behind this approach is that there is only one piece of contextual information that is contributing to the zero trafficability at any specific time.

A second red-flagging reasoner can be formulated as a hypotheses testing problem. Assuming that the hypotheses for the red-flagging reasoner problem can be stated as:

- $\mathcal{H}_1$: Track is influenced by all contextual data except depth.
- $\mathcal{H}_2$: Track is influenced by all contextual data except marked channels.
- $\mathcal{H}_3$: Track is influenced by all contextual data except anti-shipping factor.
- $\mathcal{H}_4$: Track is influenced by all contextual data except high-value unit factor.
- $\mathcal{H}_5$: Track is influenced by all contextual data.

Five different Con-Trackers can be designed according to the five different hypotheses given above. The hypothesis corresponds to the Con-Tracker that has the maximum likelihood value $p(\mathbf{y}_k \mid \mathbf{x}_k^-)$ is selected as the candidate hypothesis. The *a priori* probability density function $p(\mathbf{y}_k \mid \mathbf{x}_k^-)$ can easily be obtained from the appropriate Con-Tracker equations.

5. RESULTS

In order to evaluate the performance of the *a priori* subsystem, a test case scenario is developed where we consider Hampton Roads Bay, Virginia, near the Norfolk Naval Station. The area of interest is first divided into a 15×20 grid-field as shown in Fig. 2. Afterward, a trafficability value is assigned to each cell based on the target vessel type and the individual contextual data. Since we consider four different contextual data here, a combined trafficability value is also assigned to each cell by combining the four individual trafficability values. As shown in Fig. 2, the harbor area contains three high-value units centered around cells $(2,11)$, $(6,14)$, and $(11\ldots15,8)$. The harbor area also contains two anti-shipping areas centered about cells $(4,2)$ and $(5,17)$. There are several marked shipping lanes in the harbor area that are indicated by shaded purple channels. For simulation purposes, we consider four different target vessels.

1) Two Ski Boats: Both ski boat tracks are indicated by red lines in Fig. 2. Details on the individual ski boats are given below:
   - Ski Boat 1: Ski boat 1 starts in cell $(15,8)$ and travels toward cell $(2,1)$. Ski boat 1 crosses over two different marked channels at cells $(14,7)$ and $(11,5)$. Finally, the ski boat 1 crosses over a anti-shipping area located around cell $(14,2)$ and travels towards cell $(2,1)$.
   - Ski Boat 2: Ski boat 2 starts in cell $(15,1)$ and travels toward cell $(4,20)$. Ski boat 2 crosses over a marked channel in cell $(11,7)$ and an anti-shipping area located around cell $(5,17)$ while traveling toward cell $(4,20)$.
2) Tugboat: Tugboat starts in cell $(1,20)$ and travels along the marked channel toward cell $(13,1)$. Its track is indicated by green lines in Fig. 2.
3) Sailboat: This boat is considered as a distressed vessel that is stranded in cell $(7,3)$ due to low water depth.

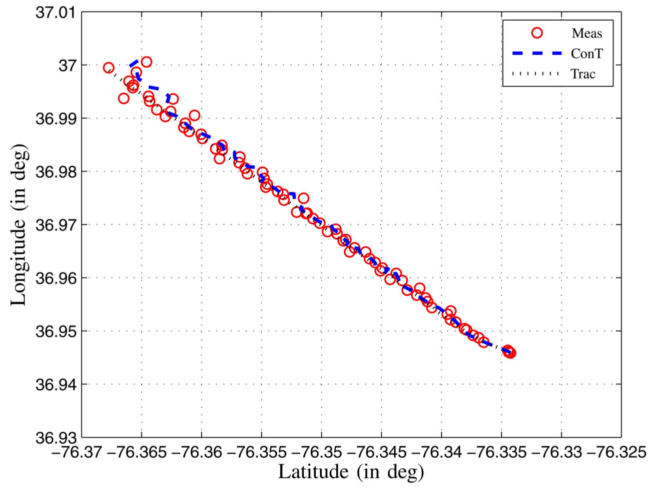


Fig. 5. Ski boat 1 trajectories: Measured position (Meas), Con-Tracker estimate (ConT) & tracker estimate (Trac).

For simulation purposes, the measurements are assumed to be obtained from an X-band coastal radar with a sampling frequency of 1/6 Hz. More details on state-of-the-art maritime surveillance technologies can be found in [48] and [47]. The measurement covariance is assumed to be of magnitude 1×10^{-7} to 2×10^{-7}. In the MMAE algorithm, 200 different filters are implemented using process noise covariance values in the range of 1×10^{-10} to 2×10^{-20}. The initial error covariance is selected to be $10^{-6} \times I_4$ and the initial process noise covariance estimate is selected to be the ensemble mean of process noise covariance values. Details of the simulation results are given next.

5.1. Ski Boat 1

As shown in Fig. 2, ski boat 1 starts in cell $(15,7)$ and travels toward cell $(2,1)$. Fig. 5 shows the measured and estimated trajectories for ski boat 1. Fig. 5 contains the estimated trajectories from both context-aided Con-Tracker (denoted as ConT) and the traditional Kalman filter-based tracker (denoted as Trac). Fig. 6(a) shows the estimated process noise covariance variance values from the Con-Tracker/MMAE $\{\hat{q}_{1_k}, \hat{q}_{2_k}\}$ and the Kalman filter tracker/MMAE $\{\breve{q}_{1_k}, \breve{q}_{2_k}\}$. The normalized process noise covariance norm, $\|q_k\|$, is given in Fig. 6(b). Note

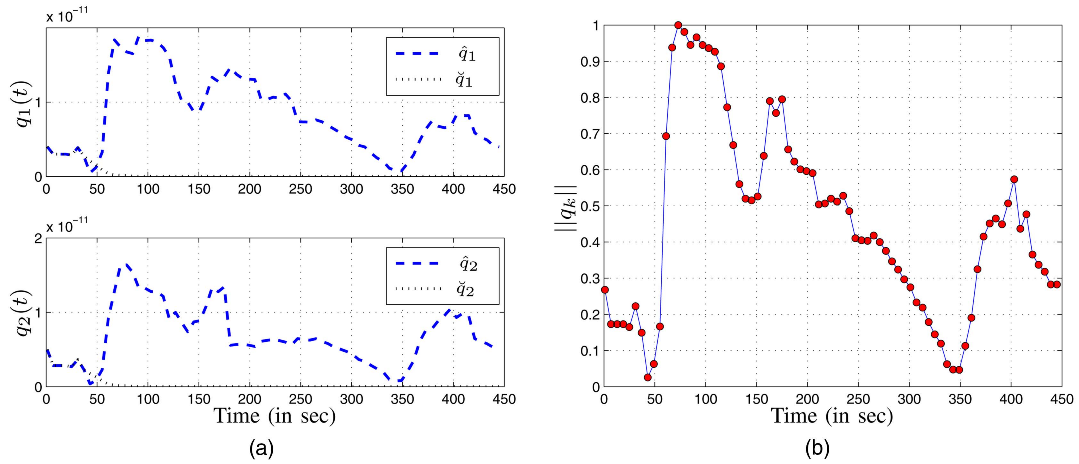


Fig. 6. Con-Tracker & tracker-estimated process noise covariance and normalized norm for ski boat 1. (a) Estimated q_1 and q_2. (b) Normalized process noise covariance norm.

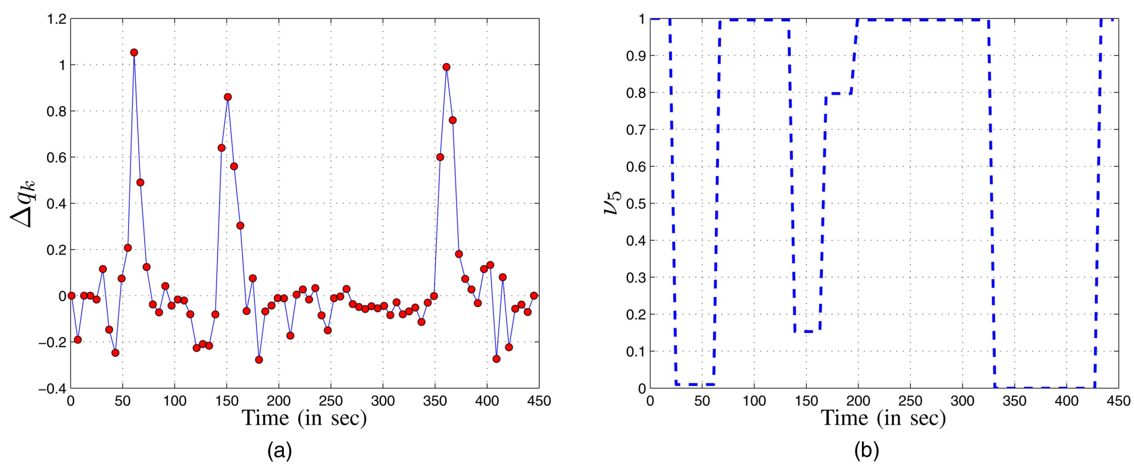


Fig. 7. Rate of change of normalized process noise covariance norm and trafficability values for ski boat 1. (a) Rate of change of $\|q_k\|$. (b) Trafficability values.

the sudden increase in $\|q_k\|$ at times 50 sec, 150 sec, and 350 sec. The first increase in the process noise covariance values occur when the ski boat crosses over the marked channel located about cell (14,7). The second increase in process noise covariance values occurs when the ski boat crosses over the second marked channel located about the cell (11,5) at around 145 sec. The final increase in the process noise covariance values occurs when the ski boat enters the anti-shipping area located about cell (4,2) at around 350 sec.

Shown in Fig. 7 are the rate of change of normalized process noise covariance norm, Δq_k, and the trafficability values, ν, for ski boat 1. The target vehicle (ski boat 1) is red-flagged based on the rate of change of normalized process noise covariance norm. The maximum allowable Δq_k is selected to be $\Delta q_{\max} = 0.8$. Note that at times 50 sec, 150 sec, and 350 sec, Δq_k is higher than its threshold value, and therefore, the target vehicle is red-flagged at these instances. Also note the low trafficability values at these instances as shown in Fig. 7(b).

Fig. 8(a) shows θ, which is the angle between the velocity vector and the local y-axis, for the Con-Tracker and the traditional Kalman filter-based tracker. The angle is measured positive clockwise and negative counterclockwise. Note that the angle obtained from the Kalman filter based tracker is much smoother compared to the one obtained from the Con-Tracker. The discrepancies in the Con-Tracker's angle is due to the velocity nudging that occurs when the target vehicle encounters a zero-trafficability area. Also note that when the boat is traveling in a completely traversable region, θ obtained for the Con-Tracker and the traditional Kalman filter-based tracker are very similar. Fig. 8(b) shows the red-flag alerts for ski boat 1. Here, zero indicates a no red-flag alert and one indicates a red-flag occurrence. Note that the red-flag occurrence and the large deviations in θ are consistent with the results shown in Fig. 7.

5.2. Ski Boat 2

As depicted in Fig. 2, the second ski boat starts in cell (15,1) and travels toward cell (4,20). Fig. 9 shows the measured and estimated tracks for ski boat 2. Fig. 10(a) contains the estimated process noise covariance variance values from the Con-Tracker/MMAE

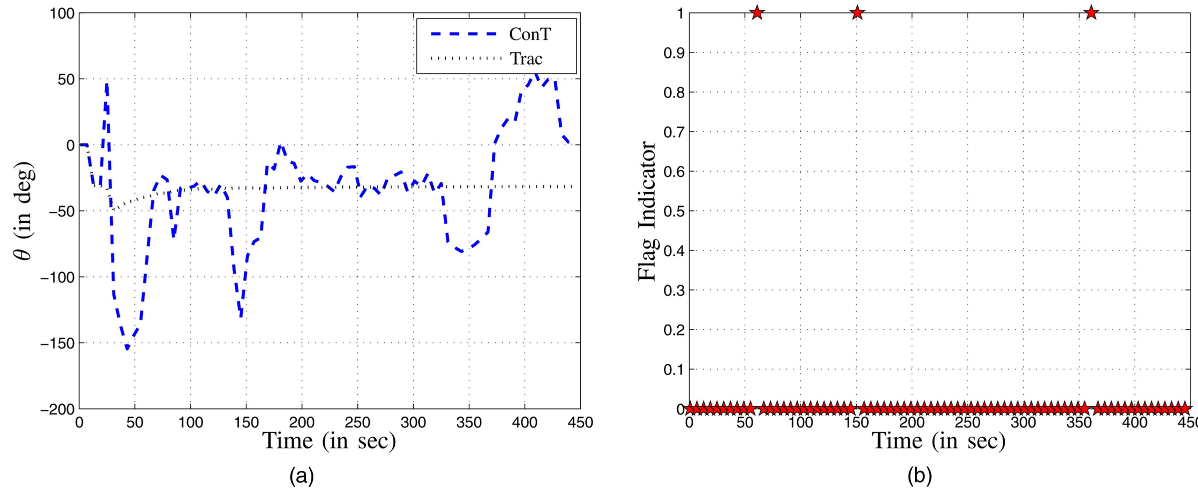


Fig. 8. Con-Tracker & tracker-estimated direction and red-flag indicator for ski boat 1. (a) Boat direction. (b) Red-flag indicator.

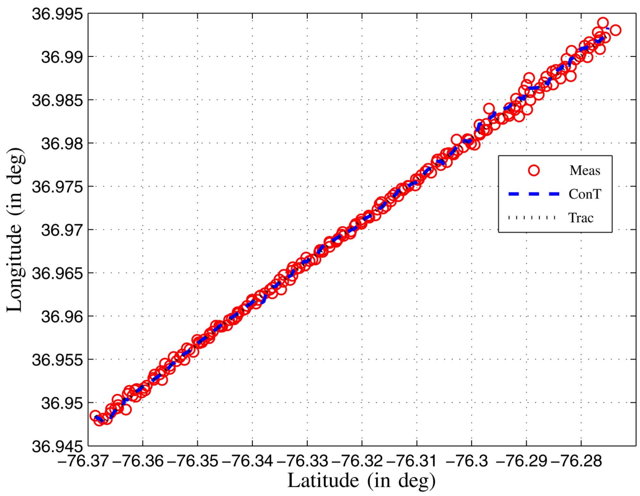


Fig. 9. Ski boat 2 trajectories: Measured position (Meas), Con-Tracker estimate (ConT) & tracker estimate (Trac).

$\{\hat{q}_{1_k}, \hat{q}_{2_k}\}$ and the Kalman filter-based tracker/MMAE $\{\breve{q}_{1_k}, \breve{q}_{2_k}\}$. Fig. 10(b) shows the normalized process noise covariance norm, $\|q_k\|$, for ski boat 2. Note the sudden increase in $\|q_k\|$ at times 400 sec and 850 sec. The first increase in the process noise covariance values occurs when ski boat 2 crosses over the marked channel located about cell $(11,7)$. The second increase in the process noise covariance occurs when ski boat 2 enters the anti-shipping area located about cell $(5,17)$ around 850 sec.

Shown in Fig. 11 are the rate of change of normalized process noise covariance norm, Δq_k, and the trafficability values, ν, for ski boat 2. The maximum allowable Δq_k for ski boat 2 is also selected to be $\Delta q_{\max} = 0.80$. Note that at times 400 sec and 850 sec, Δq_k is higher than its threshold value, and therefore, the target vehicle would be red-flagged at these instances. Also note the low trafficability values at these instances as shown in Fig. 11(b).

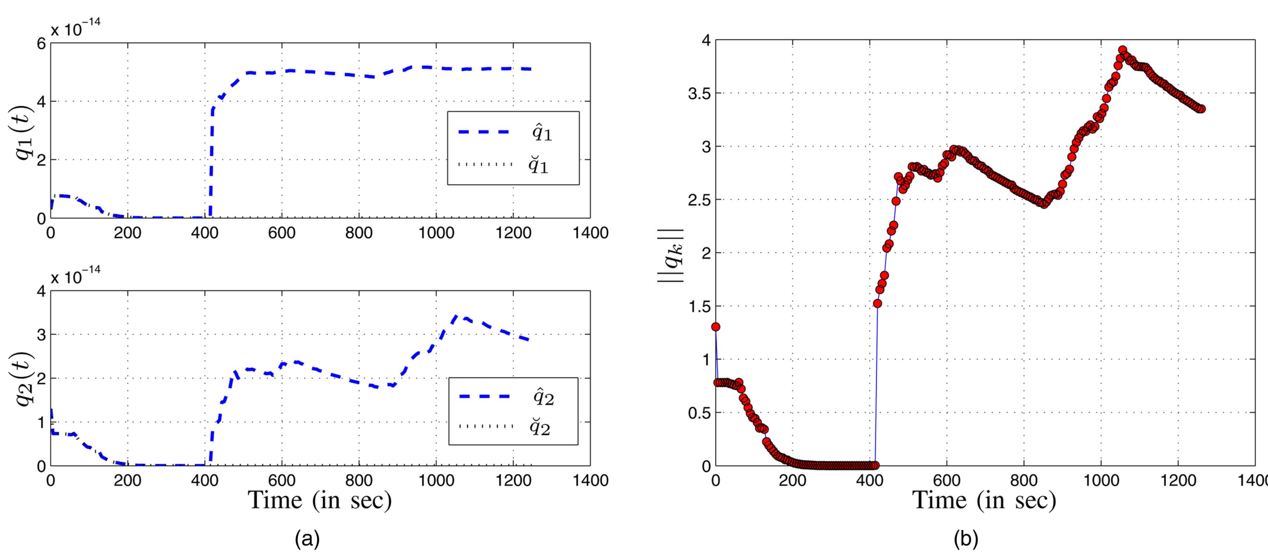
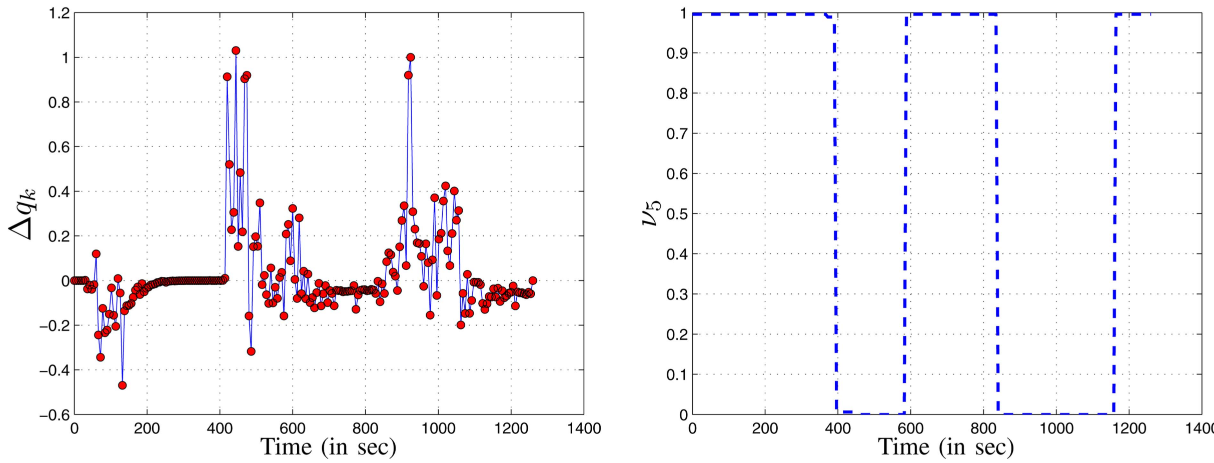
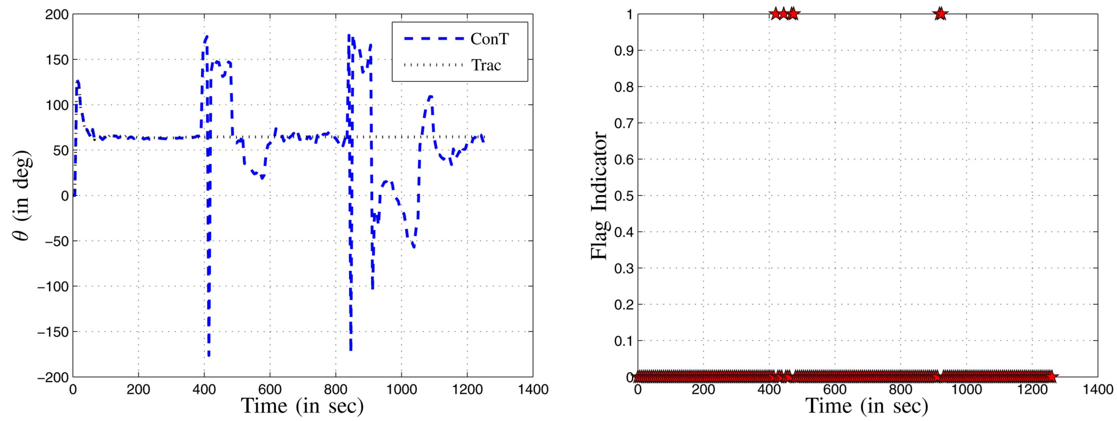


Fig. 10. Con-Tracker & tracker-Estimated process noise covariance and normalized norm for ski boat 2. (a) Estimated q_1 and q_2. (b) Normalized process noise covariance norm.



(a) (b)

Fig. 11. Rate of change of normalized process noise covariance norm and trafficability values for ski boat 2. (a) Rate of change of $\|q_k\|$. (b) Trafficability values.



(a) (b)

Fig. 12. Con-Tracker & tracker-estimated direction and red-flag indicator for ski boat 2. (a) Boat direction. (b) Red-flag indicator.

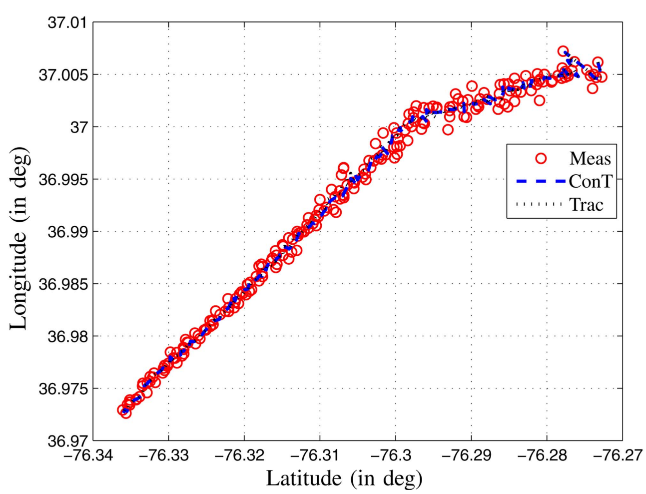


Fig. 13. Tugboat trajectories: Measured position (Meas), Con-Tracker estimate (ConT) & tracker estimate (Trac).

Fig. 12(a) shows the the angle between the velocity vector and the local y-axis, for the Con-Tracker and the Kalman filter-based tracker. Similar to the results obtained for ski boat 1, the angle obtained from the Kalman filter-based tracker is much smoother compared to the one obtained from the Con-Tracker. The discrepancies in the Con-Tracker's angle is due to the velocity nudging that occurs when the target vehicle encounters a zero-trafficability area. Fig. 12(b) shows the red-flag alerts for ski boat 2. Note that the red-flag occurrence and the large deviations in θ are consistent with the results shown in Fig. 11.

5.3 Tugboat

The tugboat starts in cell $(1,20)$ and travels along the marked channel toward cell $(13,1)$. Fig. 13 shows the measured and estimated tracks for the tugboat. Fig. 14(a) contains the estimated process noise covariance variance values from the Con-Tracker/MMAE $\{\hat{q}_{1_k}, \hat{q}_{2_k}\}$ and the Kalman filter-based tracker/MMAE $\{\breve{q}_{1_k}, \breve{q}_{2_k}\}$. Fig. 14(b) shows the normalized process noise covariance norm, $\|q_k\|$, for the tugboat. Shown in Fig. 15 are the rate of change of normalized process noise covariance norm, Δq_k, and the trafficability values, ν, for the tugboat.

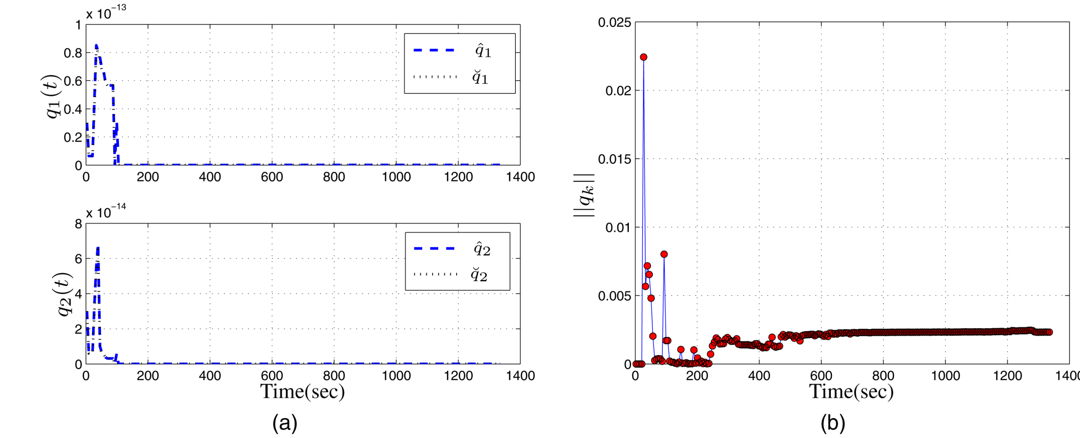


Fig. 14. Con-Tracker & tracker-estimated process noise covariance and normalized norm for tugboat. (a) Estimated q_1 and q_2. (b) Normalized process noise covariance norm.

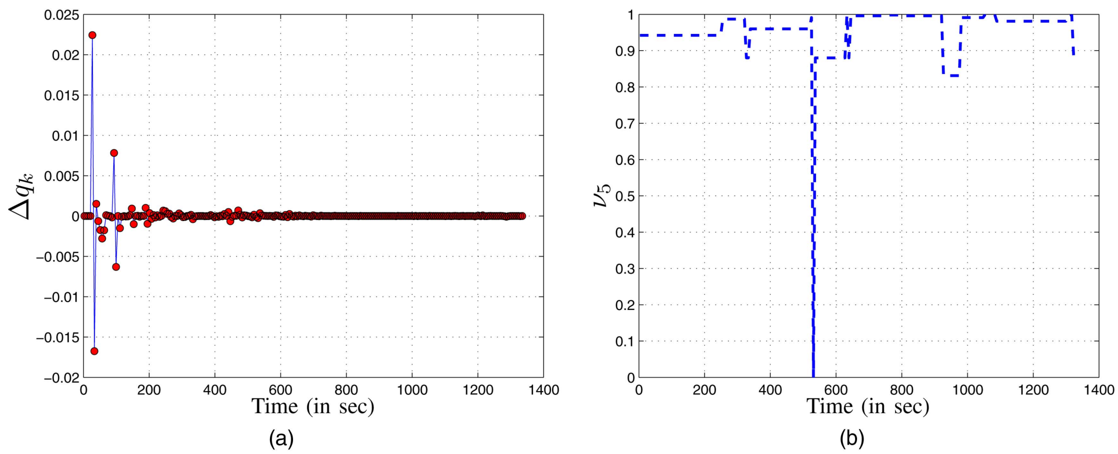


Fig. 15. Rate of change of normalized process noise covariance norm and trafficability values for tugboat. (a) Rate of change of $\|q_k\|$. (b) Trafficability values.

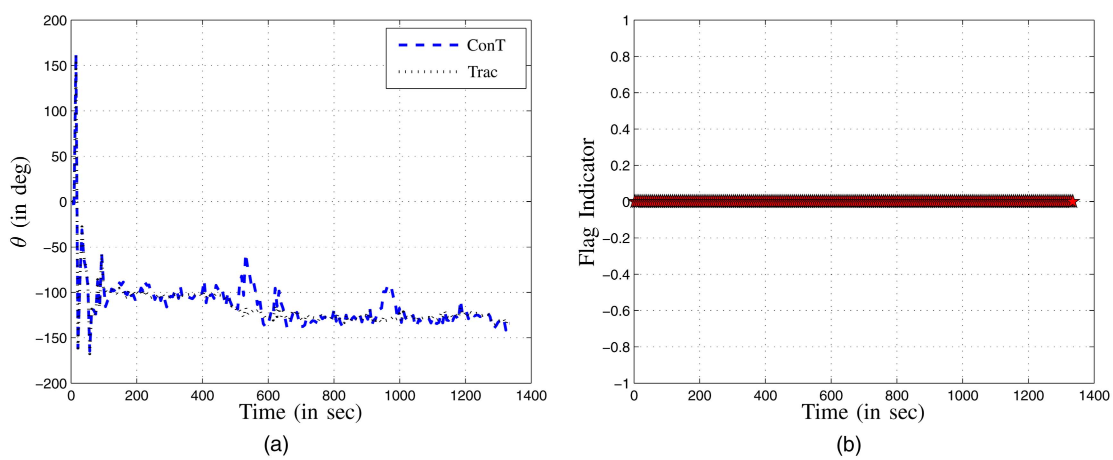


Fig. 16. Con-Tracker & tracker-estimated direction and red-flag indicator for tugboat. (a) Boat direction. (b) Red-flag indicator.

Fig. 16(a) shows the the angle between the velocity vector and the local y-axis, for the Con-Tracker and the Kalman filter-based tracker. Fig. 16(b) shows the red-flag alerts for the tugboat. Note that there is no red-flag occurrence for the tugboat since it remains in the marked shipping channel.

5.4 Sailboat

A sailboat is considered as a distressed vessel that is stranded in cell (7,3) due to low water depth. Fig. 17 shows the measured and estimated tracks for the sailboat. Fig. 18(a) contains the estimated process noise covariance variance values from the Con-Tracker/MMAE

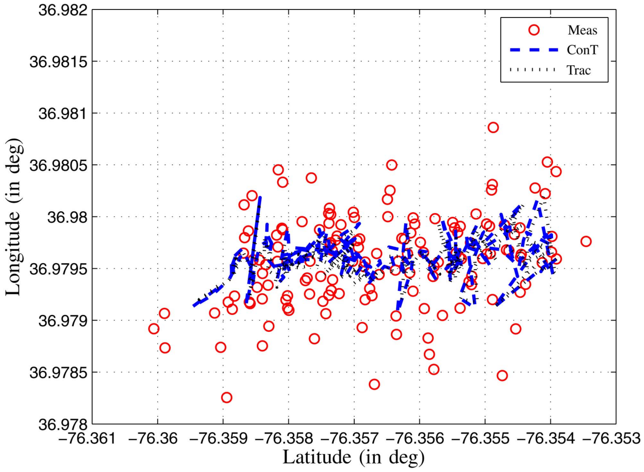


Fig. 17. Sailboat trajectories: Measured position (Meas), Con-Tracker estimate (ConT) & tracker estimate (Trac).

$\{\hat{q}_{1_k}, \hat{q}_{2_k}\}$ and the Kalman filter-based tracker/MMAE $\{\breve{q}_{1_k}, \breve{q}_{2_k}\}$. Fig. 18(b) shows the normalized process noise covariance norm, $\|q_k\|$, for the sailboat. Shown in Fig. 19 are the rate of change of normalized process noise covariance norm, Δq_k, and the trafficability values, ν, for the sailboat. The maximum allowable Δq_k for the sailboat is also selected to be $\Delta q_{\max} = 0.80$. Fig. 20(a) shows the the angle between the velocity vector and the local y-axis, for the Con-Tracker and the Kalman filter-based tracker. Fig. 20(b) shows the red-flag alerts for the sailboat. Note that the red-flag occurrences of the sailboat are consistent with the low trafficability values given in Fig. 19(b).

6. FINAL REMARKS

The objective of this work is to develop a context-aware target model and exploit available contextual information to provide a hypothesis on suspicious target maneuvers. The proposed concept involves utilizing the L1 tracking approach to perform L2/L3 situation and threat, refinement and assessment. A new context-aided tracker called the Con-Tracker is developed here. This tracker, which has its foundation in the standard Kalman filter based tracker, incorporates the available contextual information into the target vehicle model as trafficability values. Based on the trafficability values, the target

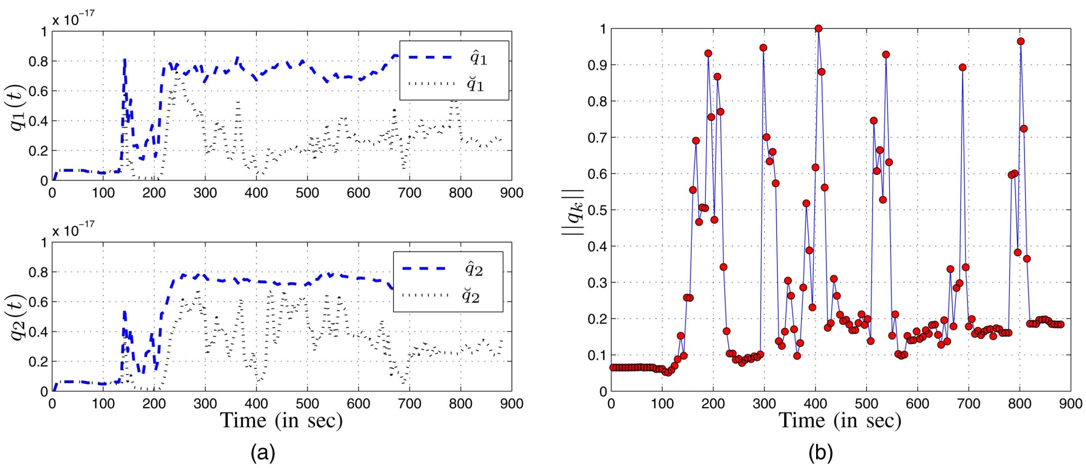


Fig. 18. Con-Tracker & tracker-estimated process noise covariance and normalized norm for sailboat. (a) Estimated q_1 and q_2. (b) Normalized process noise covariance norm.

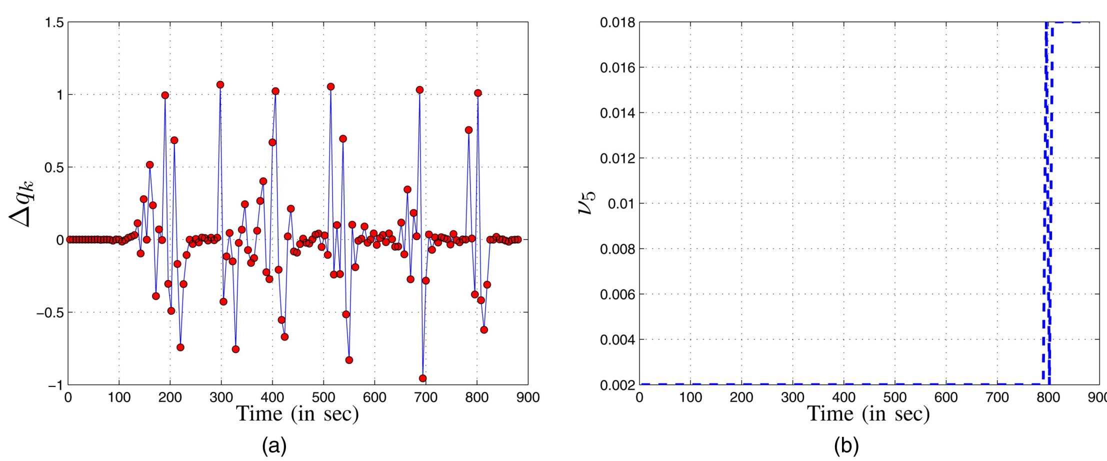


Fig. 19. Rate of change of normalized process noise covariance norm and trafficability values for sailboat. (a) Rate of change of $\|q_k\|$. (b) Trafficability values.

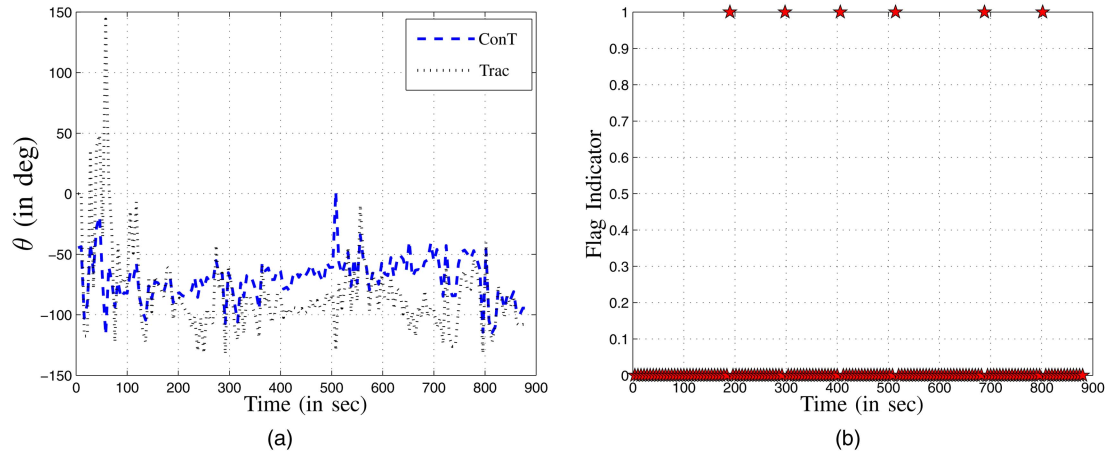


Fig. 20. Con-Tracker & tracker-estimated direction and red-flag indicator for sailboat. (a) Boat direction. (b) Red-flag indicator.

vehicle is either attracted or repelled from a particular area. Though the traditional Kalman filter-based tracker uses a near-constant velocity model, the Con-Tracker allows reasonable variations in velocity that are consistent with the contextual information. Any abrupt variations in velocity that is inconsistent with the context-aware target model would account for suspicious target maneuvers. Also, target maneuvers that are inconsistent with the given contextual information are also considered to be suspicious. Similar to the traditional Kalman filter-based tracker, the accuracy of the Con-Tracker estimates depends on the estimator parameters, such as the measurement noise covariance and the process noise covariance. While the measurement noise covariance can be readily obtained from sensor calibration, the process noise covariance value is usually treated as a tuning parameter. The proposed scheme utilizes a MMAE to estimate the process noise covariance value. Since the process noise is added to the near-constant velocity model to account for reasonable variations in velocity, target maneuvers involving large variations in velocity that are inconsistent with the contextual information would result in an increase in the estimated process noise covariance value. Based on the rate of change of the estimated process noise covariance values, an L2/L3 hypothesis generator red-flags the target vehicle. Simulation results indicate that the context-aided tracking enhances the reliability of erratic maneuver detection.

There are several parts of the proposed scheme that can be further modified and improved. The current L2/L3 hypothesis generator uses the process noise covariance estimated using the MMAE approach. One of the main drawbacks of the MMAE approach is that it requires a long convergence period. Once the process noise covariance value increases due to an erratic vehicle maneuver, the MMAE approach requires the vehicle to travel through a perfect trafficability area for a long period of time before the process noise covariance value settles back at its initial low value. The convergence properties of the MMAE can be improved by incorporating correlations between various measurement times, i.e., replacing the MMAE with the generalized MMAE (see [14]). The red-flagging design considered here depends on a threshold value for the rate of change of the normalized process noise covariance norm. A probabilistic red-flagging scheme, which integrates the current *posterior* error covariance and estimated process noise covariance, may be considered for future work. Finally, the accuracy and performance of the proposed scheme can be improved by considering more refined trafficability grid-field and more frequent and accurate measurements.

ACKNOWLEDGEMENT

This work was supported by Silver Bullet Solutions through an Office of Naval Research grant. The authors wish to thank David McDaniel and Todd Kingsbury from Silver Bullet Solutions for their support and generation of the data that was used in this paper.

REFERENCES

[1] A. Alouani and W. Blair
Use of a kinematic constraint in tracking constant speed, maneuvering targets.
In *Proceedings of the 30th IEEE Conference on Decision and Control*, vol. 2, 1991, 2055–2058.

[2] A. Alouani and W. Blair
Use of a kinematic constraint in tracking constant speed, maneuvering targets.
IEEE Transactions on Automatic Control, **38**, 7 (July 1993), 1107–1111.

[3] A. Alouani, W. Blair, and G. Watson
Bias and observability analysis of target tracking filters using a kinematic constraint.
In *Proceedings of the Twenty-Third Southeastern Symposium on System Theory*, 1991, 229–232.

[4] B. Anderson and J. B. Moore
Optimal Filtering.
Mineola, NY: Dover Publications, 1979.

[5] B. Anderson, J. B. Moore, and R. M. Hawkes
Model approximations via prediction error identification.
Automatica, **14**, 6 (Nov. 1978), 615–622.

[6] Y. Bar-Shalom, K. Chang, and H. Blom
Tracking a maneuvering target using input estimation versus the interacting multiple model algorithm.
IEEE Transactions on Aerospace and Electronic Systems, **25**, 2 (Mar. 1989), 296–300.

[7] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation.
New York, NY: John Wiley & Sons, Inc., 2001.

[8] J. Barker, R. Green, P. Thomas, G. Brown, and D. Salmond
Information fusion based decision support via hidden Markov models and time series anomaly detection.
In *12th International Conference on Information Fusion*, 2009, 764–771.

[9] H. A. P. Blom
An efficient filter for abruptly changing systems.
In *The 23rd IEEE Conference on Decision and Control*, Dec. 1984, 656–658.

[10] H. Blom and Y. Bar-Shalom
The interacting multiple model algorithm for systems with Markovian switching coefficients.
IEEE Transactions on Automatic Control, **33**, 8 (Aug. 1988), 780–783.

[11] R. G. Brown and P. Hwang
Introduction to Random Signals and Applied Kalman Filtering, 3rd ed.
New York, NY: John Wiley & Sons, 1997, 353–361.

[12] B. Brumback and M. Srinath
A chi-square test for fault-detection in kalman filters.
IEEE Transactions on Automatic Control, **32**, 6 (June 1987), 552–554.

[13] V. Chandola, A. Banerjee, and V. Kumar
Anomaly detection: A survey.
ACM Computing Surveys (CSUR), **41**, 3 (2009), 1–58.

[14] J. L. Crassidis and Y. Cheng
Generalized multiple-model adaptive estimation using an autocorrelation approach.
In *Proceedings of the 9th International Conference on Information Fusion*, Florence, Italy, July 2006, paper 223.

[15] J. L. Crassidis and J. L. Junkins
Optimal Estimation of Dynamic Systems.
Boca Raton, FL: CRC Press, 2004.

[16] E. Daeipour and Y. Bar-Shalom
An interacting multiple model approach for target tracking with glint noise.
IEEE Transactions on Aerospace and Electronic Systems, **31**, 2 (Apr. 1995), 706–715.

[17] M. Efe and D. Atherton
Maneuvering target tracking using adaptive turn rate models in the interacting multiple model algorithm.
In *Proceedings of the 35th IEEE Conference on Decision and Control*, vol. 3, 1996, 3151–3156.

[18] R. Fitzgerald
Simple tracking filters: Position and velocity measurements.
IEEE Transactions on Aerospace and Electronic Systems, **AES-18**, 5 (Sept. 1982), 531–537.

[19] A. M. Fosbury, T. Singh, J. L. Crassidis, and C. Springen
Ground target tracking using terrain information.
10th International Conference on Information Fusion, July 2007.

[20] J. M. Hammersley
Monte Carlo methods for solving multivariable problems.
Annals of the New York Academy of Sciences, **86** (1960), 844–874.

[21] G. Jackson, L. Lewis, and J. Buford
Behavioral anomaly detection in dynamic self-organizing systems: A cognitive fusion approach.
In *International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, Apr. 2005, 254–259.

[22] P. R. Kalata
α-β; target tracking systems: A survey.
In *American Control Conference*, 1992, 832–836.

[23] K. Kastella and C. Kreucher
Multiple model nonlinear filtering for low signal ground target applications.
IEEE Transactions on Aerospace and Electronic Systems, **41**, 2 (Apr. 2005), 549–564.

[24] O. Kessler and F. White
Data fusion perspectives and its role in information processing.
In M. E. Liggins, D. L. Hall, and J. Llinas (Eds.), *Handbook of Multisensor Data Fusion: Theory and Practice*, Boca Raton, FL: CRC Press, 2009, ch. 2.

[25] T. Kirubarajan, Y. Bar-Shalom, and K. Pattipati
Topography-based vs-imm estimator for large-scale ground target tracking.
In *IEE Colloquium on Target Tracking: Algorithms and Applications*, 1999.

[26] T. Kirubarajan, Y. Bar-Shalom, K. Pattipati, and I. Kadar
Ground target tracking with variable structure imm estimator.
IEEE Transactions on Aerospace and Electronic Systems, **36**, 1 (Jan. 2000), 26–46.

[27] T. Kirubarajan, Y. Bar-Shalom, K. Pattipati, I. Kadar, B. Abrams, and E. Eadan
Tracking ground targets with road constraints using an imm estimator.
In *Proceedings of IEEE Aerospace Conference*, **5**, 21–28 (1998), 5–12.

[28] D. Lerro and Y. Bar-Shalom
Interacting multiple model tracking with target amplitude feature.
IEEE Transactions on Aerospace and Electronic Systems, **29**, 2 (Apr. 1993), 494–509.

[29] X. Li and Y. Bar-Shalom
Design of an interacting multiple model algorithm for air traffic control tracking.
IEEE Transactions on Control Systems Technology, **1**, 3 (Sept. 1993), 186–194.

[30] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan
Interacting multiple model methods in target tracking: A survey.
IEEE Transactions on Aerospace and Electronic Systems, **34**, 1 (Jan. 1998), 103–123.

[31] R. Moose, H. Vanlandingham, and D. Mccabe
Modeling and estimation for tracking maneuvering targets.
IEEE Transactions on Aerospace and Electronic Systems, **AES-15**, 3 (May 1979), 448–456.

[32] A. Munir and D. Atherton
Maneuvering target tracking using an adaptive interacting multiple model algorithm.
In *American Control Conference*, **2** (1994), 1324–1328.

[33] A. Munir and D. Atherton
Adaptive interacting multiple model algorithm for tracking a manoeuvring target.
IEE Proceedings—Radar, Sonar and Navigation, **142**, 1 (Feb. 1995), 11–17.

[34] P. O. Nougues and D. E. Brown
We know where you are going: Tracking objects in terrain.
IMA Journal of Mathematics Applied in Business & Industry, **8** (1997), 39–58.

[35] J. Ouyang, N. Patel, and I. Sethi
Chi-square test based decision trees induction in distributed environment.
In *IEEE International Conference on Data Mining Workshops*, 15–19 (2008), 477–485.

[36] D. B. Reid and R. G. Bryson
A non-Gaussian filter for tracking targets moving over terrain.
In *Proceedings of the 12th Annual Asilomar Conference on Circuits, Systems, and Computers*, Pacific Grove, CA, Nov. 1978, 112–116.

[37] B. Ristic, B. La Scala, M. Morelande, and N. Gordon
Statistical analysis of motion patterns in ais data: Anomaly detection and motion prediction.
In *2008 11th International Conference on Information Fusion*, 2008.

[38] M. Roemer, G. Kacprzynski, and M. Schoeller
Improved diagnostic and prognostic assessments using health management information fusion.
In *IEEE Systems Readiness Technology Conference AUTOTESTCON Proceedings*, 2001, 365–377.

[39] B. Salem and T. Karim
Context-based profiling for anomaly intrusion detection with diagnosis.
In *Third International Conference on Availability, Reliability and Security*, 2008, 618–623.

[40] R. Singer
Estimating optimal tracking filter performance for manned maneuvering targets.
IEEE Transactions on Aerospace and Electronic Systems, **AES-6**, 4 (July 1970), 473–483.

[41] S. Singh, H. Tu, W. Donat, K. Pattipati, and P. Willett
Anomaly detection via feature-aided tracking and hidden Markov models.
IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, **39**, 1 (Jan. 2009), 144–159.

[42] R. F. Stengel
Optimal Control and Estimation.
New York, NY: Dover Publications, 1994, 402–407.

[43] D. Sworder and R. Hutchins
Maneuver estimation using measurements of orientation.
IEEE Transactions on Aerospace and Electronic Systems, **26**, 4 (July 1990), 625–638.

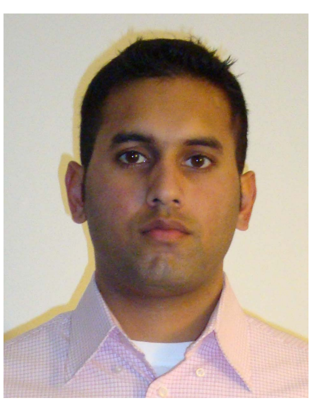
[44] D. Tenne and B. Pitman
Tracking a convoy of ground vehicles.
Center for Multisource Information Fusion, The State University of New York at Buffalo, Amherst, NY, Tech. Rep., Jan. 2002.

[45] D. Tenne, B. Pitman, T. Singh, and J. Llinas
Velocity field based tracking of ground vehicles.
Center for Multisource Information Fusion, The State University of New York at Buffalo, Amherst, NY, Tech. Rep., Aug. 2003.

[46] M. Ulmke and W. Koch
Road-map assisted ground moving target tracking.
IEEE Transactions on Aerospace and Electronic Systems, **42**, 4 (Oct. 2006), 1264–1274.

[47] M. Vespe, M. Sciotti, and G. Battistello
Multi-sensor autonomous tracking for maritime surveillance.
In *International Conference on Radar*, 2008, 525–530.

[48] M. Vespe, M. Sciotti, F. Burro, G. Battistello, and S. Sorge
Maritime multi-sensor data association based on geographic and navigational knowledge.
In *IEEE Radar Conference*, 2008.



Jemin George received his B.S. ('05), M.S. ('07), and Ph.D. ('10) in aerospace engineering from the State University of New York at Buffalo. In 2008, he was a summer research scholar with the U.S. Air Force Research Laboratory's Space Vehicles Directorate at Kirtland Air Force Base in Albuquerque, New Mexico.

He was a National Aeronautics and Space Administration Langley Aerospace Research Summer Scholar with the Langley Research Center in 2009. From 2009–2010 he was a research fellow with the Stochastic Research Group, Department of Mathematics, Technische Universitat Darmstadt, Darmstadt, Germany. He is currently with the Networked Sensing & Fusion Branch of the U.S. Army Research Laboratory.

His principal research interests include stochastic systems, control theory, nonlinear filtering, information fusion, and target tracking. He is a student member of IEEE and AIAA.

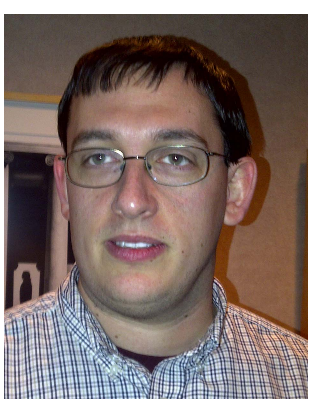
John L. Crassidis is a Professor of Mechanical and Aerospace Engineering at the University at Buffalo (UB), State University of New York, and also associate director for UB's Center for Multisource Information Fusion. He received his B.S., M.S., and Ph.D. in mechanical engineering from the State University of New York at Buffalo. Prior to joining UB in 2001, he held previous academic appointments at Catholic University of America from 1996 to 1998 and Texas A&M University from 1998 to 2001. From 1996 to 1998, he was a NASA postdoctoral research fellow at Goddard Space Flight Center, where he worked on a number of spacecraft projects and research ventures involving attitude determination and control systems.

His current research interests include nonlinear estimation and control theory, spacecraft attitude determination and control, attitude dynamics and kinematics, and robust vibration suppression. He is first author of the textbook *Optimal Estimation of Dynamic Systems* and has authored or coauthored more than 150 journal and refereed conference papers.

Tarunraj Singh received his B.E. degree from Bangalore University, Bangalore, India, his M.E. degree from the Indian Institute of Science, Bangalore, and Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, all in mechanical engineering. He was a postdoctoral fellow with the Aerospace Engineering Department, Texas A&M University, College Station.

Since 1993, he has been with the University at Buffalo, Buffalo, NY, where he is currently a professor in the Department of Mechanical and Aerospace Engineering. He was a von Humboldt Fellow and spent his sabbatical at the Technische Universitat Darmstadt, Darmstadt, Germany, and at the IBM Almaden Research Center in 2000–2001. He was a National Aeronautics and Space Administration Summer Faculty Fellow with the Goddard Space Flight Center in 2003. His research has been supported by the National Science Foundation, Air Force Office of Scientific Research, National Security Agency, Office of Naval Research, and various industries, including MOOG Inc. Praxair and Delphi Thermal Systems.

He has published more than 175 refereed journal and conference papers and has presented over 40 invited seminars at various universities and research laboratories. His research interests are in robust vibration control, estimation, and intelligent transportation.



Adam M. Fosbury received his B.S. in aerospace engineering and M.S. and Ph.D. in mechanical engineering from the State University of New York at Buffalo.

From 2006 to 2007, he was a National Research Council Postdoctoral Research Fellow with the Air Force Research Laboratory Space Vehicles Directorate. In 2007, he became a full-time employee of the Air Force Research Laboratory. He left AFRL for the Johns Hopkins Applied Physics Laboratory in 2010 and is now performing guidance, navigation and control work for several spacecraft.

He is currently a member of the AIAA Technical Committee on Guidance, Navigation and Control.

Multi-Stage Multiple-Hypothesis Tracking

STEFANO CORALUPPI
CRAIG CARTHEL

While multiple hypothesis tracking (MHT) is widely acknowledged as an effective methodology for multi-target surveillance, there is a challenge to manage effectively a potentially large number of track hypotheses. Advanced single-stage *track-while-fuse* does not always offer the best processing scheme. We study two instances where multi-stage MHT processing is beneficial—dense target scenarios and complementary-sensor surveillance—and propose two processing schemes for these challenges: *track-break-fuse* and *track-before-fuse*, respectively. We provide simulation results demonstrating the advantages of these schemes over *track-while-fuse*. More generally, we argue that multi-stage MHT offers a powerful and flexible paradigm to circumvent limitations in conventional MHT processing.

Manuscript received October 26, 2009; revised December 21, 2010 and April 8, 2011; released for publication April 21, 2011.

Refereeing of this contribution was handled by Dale Blair.

Authors' address: Compunetix Inc., 2420 Mosside Boulevard, Monroeville PA 15146, Email: {stefano.coraluppi,craig.carthel}@compunetix.com.

1557-6418/11/$17.00 © 2011 JAIF

1. INTRODUCTION

A broad overview of approaches to data fusion is provided in [1]. The most powerful current approach to real-time, scan-based data fusion is multi-hypothesis tracking (MHT), which was first introduced in the late 1970s [11] and made feasible in the mid-1980s with the track-oriented approach [9]. A number of enhancements to the basic approach have appeared over the years [1].

If contact measurement information is available at the tracker output, one can think of a multi-target tracker as a *filter* of sorts that discards spurious contacts and associates the remaining ones through track labeling. As such, tracking is a modular operator which, when applied to contact-level data, takes as input singleton (i.e. single-measurement) tracks. More generally, a mix of track-level and contact-level feeds may be provided to the tracker. Upstream track labels are preserved in downstream processing, except in cases where discrepancies are detected in downstream tracking. This tracker modularity allows for arbitrarily complex multi-stage data fusion architectures. This philosophy, combined with the necessary software modularity, is the basis for the multi-stage MHT approach that we consider in this paper. We find that in some applications multi-stage MHT processing outperforms single-stage MHT processing.

In this paper, we introduce two multi-stage MHT architectures and compare these to single-stage, *track-while-fuse* processing. The first multi-stage architecture, *track-break-fuse*, is computationally efficient without sacrificing the tracking performance of *track-while-fuse*. The second architecture, *track-before-fuse*, provides further computational efficiency at the cost of some tracking performance. The *track-while-fuse* approach is intractable when the application requires deep hypothesis trees; conversely, both of the multi-stage MHT approaches that we introduce here identify a small set of relevant association hypotheses, enabling deep hypothesis trees.

The paper is organized as follows. In Section 2, we provide a short introduction to standard (*track-while-fuse*) track-oriented MHT, following closely on the formalism introduced in [9]. The multi-stage MHT architectures of interest, *track-break-fuse* and *track-before-fuse*, are introduced in Section 3. In Section 4 we study *track-break-fuse* for a challenging, slowly-crossing targets problem. In Section 5 we study *track-before-fuse* for multi-sensor surveillance with complementary, multi-scale sensors. Concluding remarks are in Section 6.

Early results on the multi-stage processing introduced here are in [6] (*track-break-fuse*) and [3] (*track-before-fuse*). A related MHT approach to *track-before-fuse* is discussed in [4], which introduces group-tracking logic to enable deep hypothesis trees. Additionally, within the MHT framework, some techniques to hypothesis management do exist, including K-best assignment or hypothesis-clustering approaches [7, 10]. However, the difficulty is that one must maintain *relevant* track hypotheses for significant time duration, and often the top-scoring global hypotheses do not have sufficient diversity to insure that this is achieved. Hypothesis clustering may only partially ameliorate the situation. Possibly, one might adopt a probabilistic data association framework to address the problem, leveraging the approach introduced in [8].

2. TRACK-ORIENTED MULTIPLE HYPOTHESIS TRACKING

A key challenge in multi-sensor multi-target tracking is *measurement origin uncertainty*. That is, unlike a classical nonlinear filtering problem, we do not know how many objects are in the surveillance region, and which measurements are to be associated. New objects may be born in any given scan, and existing objects may die.

We assume that for each sensor scan, contact-level (or detection-level) data are available, in the sense that signal processing techniques are applied to raw sensor data yielding contacts for which the detection and localization statistics are known. We are interested in a scan-based (or real-time) approach that, perhaps with some delay, yields an estimate of the number of objects and corresponding object state estimates at any time.

Several approaches to contact-level scan-based tracking exist. In this section, we employ a hybrid-state formalism to describe the track-oriented multiple-hypothesis tracking approach. Our approach follows closely the one introduced in [9]. We assume Poisson distributed births at each scan with mean λ_b, Poisson distributed false returns with mean λ_{fa}, object detection probability p_d, object death or termination probability p_χ at each scan. (We neglect the *time-dependent* nature of birth and death probabilities as would ensue from an underlying continuous-time formulation, and we neglect as well inter-scan birth and death events.)

We have a sequence of sets of contacts $Z^k = (Z_1,\ldots, Z_k)$, and we wish to estimate the state history X^k for all objects present in the surveillance region. X^k is compact notation that represents the state trajectories of targets that exist over the time sequence $(t_1,\ldots,t_k)$. Note that each target may exist for a subset of these times, with a single birth and a single death occurrence i.e. targets do not reappear. We introduce the auxiliary discrete state history q^k that represents a full interpretation of all contact data: which contacts are false, how the object-originated ones are to be associated, and when objects are born and die. There are two fundamental assumptions of note. The first is that there are no target births in the absence of a corresponding detection, i.e. we do not reason over new, undetected objects. The second is that there is *at most* one contact per object per scan.

We are interested in the probability distribution $p(X^k \mid Z^k)$ for object state histories given data. This quantity can be obtained by conditioning over all possible auxiliary states histories q^k.

$$p(X^k \mid Z^k) = \sum_{q^k} p(X^k, q^k \mid Z^k)$$
$$= \sum_{q^k} p(X^k \mid Z^k, q^k) p(q^k \mid Z^k). \quad (1)$$

A pure MMSE approach would yield the following:

$$\hat{X}_{\text{MMSE}}(Z^k) = E[X^k \mid Z^k]$$
$$= \sum_{q^k} E[X^k \mid Z^k, q^k] p(q^k \mid Z^k). \quad (2)$$

The track-oriented MHT approach is a mixed MMSE/MAP one, whereby we identify the MAP estimate for the auxiliary state history q^k, and identify the corresponding MMSE estimate for the object state history X^k conditioned on the estimate for q^k.

$$\hat{X}(Z^k) = \hat{X}_{\text{MMSE}}(Z^k, \hat{q}^k) \quad (3)$$

$$\hat{q}^k = \hat{q}_{\text{MAP}}(Z^k) = \arg\max_{q^k} p(q^k \mid Z^k). \quad (4)$$

Each feasible q^k corresponds to a global hypothesis. (The set of global hypotheses is generally constrained via measurement gating and hypothesis generation logic.) We are interested in a *recursive* and *computationally efficient* expression for $p(q^k \mid Z^k)$ that lends itself to functional optimization without the need for explicit enumeration of global hypotheses. We do so through repeated use of Bayes' rule. Note that, for notational simplicity, we use $p(\cdot)$ for both probability density and probability mass functions. Also, as discussed in Section 2, the normalizing constant c_k does not impact MAP estimation.

$$p(q^k \mid Z^k) = \frac{p(Z_k \mid Z^{k-1}, q^k) p(q^k \mid Z^{k-1})}{c_k}$$
$$= \frac{p(Z_k \mid Z^{k-1}, q^k) p(q_k \mid Z^{k-1}, q^{k-1}) p(q^{k-1} \mid Z^{k-1})}{c_k} \quad (5)$$

$$c_k = p(Z_k \mid Z^{k-1})$$
$$= \sum_{q^k} p(Z_k \mid Z^{k-1}, q^k) p(q^k \mid Z^{k-1}). \quad (6)$$

Recall that we assume that in each scan the number of target births is Poisson distributed with mean λ_b, the number of false returns is Poisson distributed with mean λ_{fa}, targets die with probability p_χ, and targets are detected with probability p_d. The recursive expression (5) involves two factors that we consider in turn, with the discrete state probability one first. It will be useful to introduce the aggregate variable ψ_k (consistent with the approach in [9]) that accounts for the number of detections d for the τ existing tracks, the number of track deaths χ, the number of new tracks b, and the number of false returns $r-d-b$, where r is the number of contacts in the current scan.

$$p(q_k \mid Z^{k-1},q^{k-1}) = p(\psi_k \mid Z^{k-1},q^{k-1})p(q_k \mid Z^{k-1},q^{k-1},\psi_k) \tag{7}$$

$$p(\psi_k \mid Z^{k-1},q^{k-1}) = \left\{\binom{\tau}{\chi} p_\chi^\chi (1-p_\chi)^{\tau-\chi}\right\} \cdot \left\{\binom{\tau-\chi}{d} p_d^d (1-p_d)^{\tau-\chi-d}\right\} \cdot \left\{\frac{\exp(-\lambda_b)p_d^b \lambda_b^b}{b!}\right\} \cdot \left\{\frac{\exp(-\lambda_{fa})\lambda_{fa}^{r-d-b}}{(r-d-b)!}\right\} \tag{8}$$

$$p(q_k \mid Z^{k-1},q^{k-1},\psi_k) = \frac{1}{\binom{\tau}{\chi}\binom{\tau-\chi}{d}\left(\frac{r!}{(r-d)!}\right)\binom{r-d}{b}}. \tag{9}$$

Substituting (8–9) into (7) and simplifying yields the following.

$$p(q_k \mid Z^{k-1},q^{k-1}) = \left\{\frac{\exp(-\lambda_b-\lambda_{fa})\lambda_{fa}^r}{r!}\right\} p_\chi^\chi((1-p_\chi)(1-p_d))^{\tau-\chi-d}\left(\frac{(1-p_\chi)p_d}{\lambda_{fa}}\right)^d \left(\frac{p_d\lambda_b}{\lambda_{fa}}\right)^b. \tag{10}$$

The first factor in (5) is given below, where $Z_k = \{z_j, 1 \le j \le r\}$, $|J_d| + |J_b| + |J_{fa}| = r$, and the factors on the R.H.S. are derived from filter innovations, filter initiations, and the false contact distribution (generally uniform over measurement space). For example, in the linear Gaussian case, $f_d(z_j \mid Z^{k-1},q^k)$ is a Gaussian residual, i.e. it is the probability of observing z_j given a sequence of preceding measurements. If there is no prior information on the target, $f_b(z_j \mid Z^{k-1},q^k)$ is generally the value of the uniform density function over measurement space. Similarly, $f_{fa}(z_j \mid Z^{k-1},q^k)$ is as well usually taken to be the value of the uniform density function over measurement space, under the assumption of uniformly distributed false returns. Note that the expressions given here are general and allow for quite general target and sensor models.

$$p(Z_k \mid Z^{k-1},q^k) = \prod_{j\in J_d} f_d(z_j \mid Z^{k-1},q^k) \cdot \prod_{j\in J_b} f_b(z_j \mid Z^{k-1},q^k) \cdot \prod_{j\in J_{fa}} f_{fa}(z_j \mid Z^{k-1},q^k). \tag{11}$$

Substituting (10–11) into (5) and simplifying results in (12–13). This expression is the key enabler of track-oriented MHT. In particular, it provides a recursive expression for $p(q^k \mid Z^k)$ that consists of a number of factors that relate to its constituent local track hypotheses.

An implicit reduction in the set of hypotheses in (12–13) is that target births are assumed to occur only in the presence of a detection (i.e. there is no reasoning over un-detected births). Correspondingly, the factor p_d reduces the effective birth rate to $p_d\lambda_b$ (though surprisingly the factor is absent in [9]). Further, in the first scan of data, it would be appropriate to replace $p_d\lambda_b$ by $p_d\lambda_b/p_\chi$ to account properly for the steady-state expected number of targets. (More generally, target birth and death parameters should reflect sensor scan rates, as the underlying target process is defined in continuous time.) Further reduction in the set of hypotheses is generally achieved via *measurement gating* procedures [1]. Finally, for a given track hypothesis, one usually applies rule-based spawning of a missed detection or termination hypothesis, but not both (e.g. only spawn a missed detection hypothesis until a sufficiently-long sequence of missed detection is reached).

One cannot consider too large a set of scans before pruning or merging local (or track) hypotheses in some fashion. A popular mechanism to control these hypotheses is *n-scan pruning*. This amounts to solving (4), generally by a relaxation approach to an integer programming problem [4, 6, 11], followed by pruning of all local hypotheses that differ from $\hat{q}^k$ in the first scan. This methodology is applied after each new scan

$$p(q^k \mid Z^k) = p_\chi^\chi((1-p_\chi)(1-p_d))^{\tau-\chi-d}\prod_{j\in J_d}\left[\frac{(1-p_\chi)p_d f_d(z_j \mid Z^{k-1},q^k)}{\lambda_{fa}f_{fa}(z_j \mid Z^{k-1},q^k)}\right]\prod_{j\in J_b}\left[\frac{p_d\lambda_b f_b(z_j \mid Z^{k-1},q^k)}{\lambda_{fa}f_{fa}(z_j \mid Z^{k-1},q^k)}\right]\frac{p(q^{k-1} \mid Z^{k-1})}{\bar{c}_k} \tag{12}$$

$$\bar{c}_k = \frac{c_k}{\left\{\frac{\exp(-\lambda_b-\lambda_{fa})}{r!}\lambda_{fa}^r\right\}\prod_{j\in J_d\cup J_b\cup J_{fa}} f_{fa}(z_j \mid Z^{k-1},q^k)}. \tag{13}$$

of data are received, resulting in a fixed-delay solution to the tracking problem.

Often, *n-scan* pruning is referred to as a *maximum likelihood* (ML) approach to hypothesis management. ML estimation is closely related to *maximum a posteriori* (MAP) estimation. In particular, we have:

$$\hat{X}_{\mathrm{MAP}}(y) = \arg\max f(y \mid X) f(X) \tag{14}$$

$$\hat{X}_{\mathrm{ML}}(y) = \arg\max f(y \mid X). \tag{15}$$

Note that ML estimation is a *non-Bayesian* approach as it does not rely on a prior distribution on X. ML estimation can be interpreted as MAP estimation with a uniform prior. In the track-oriented MHT setting, n-scan pruning relies on a single parent global hypothesis, thus the ML and MAP interpretations are both valid.

Once hypotheses are resolved, in principle one has a state of object histories given by $\hat{X}(Z^k)$. In practice, it is common to apply track confirmation and termination logic to all object histories [1]. A justification for this is that it provides a mechanism to remove spurious tracks induced by the sub-optimality inherent in practical MHT implementations that include limited hypothesis generation and hypothesis pruning or merging.

Given the need for post-association track confirmation and termination logic, a reasonable simplification that is pursued in [5] is to employ equality constraints in the data-association process, which amounts to accounting for all contact data in the resolved tracks. Spurious tracks are subsequently removed in the track-extraction stage.

3. MULTI-STAGE MHT

Multi-stage fusion as performed here has two defining characteristics that differ from many legacy systems that exist today [1]. The first is that each tracker module retains measurement-level information at the output. That is, each module performs the following: it removes large numbers of measurement data, and associates the remaining measurements to form tracks over time. If the tracker is working well and the data are of reasonable quality, false measurements will largely be removed and target-originated measurements will mostly be maintained and associated into tracks that persist over time with limited fragmentation. Since measurement data are available at the tracker output, optimal track fusion and state estimation is achievable in downstage tracker modules; the cost to achieve this performance benefit is a slightly larger bandwidth requirement between processing stages. The second defining characteristic is that track fusion is achieved in a real-time, scan-based manner. Often, track fusion is performed in a post-processing batch mode that is not readily amenable to real-time surveillance application [1].

The theoretical optimality of unified, batch and centralized approaches to fusion and tracking (*track-while-fuse*) is at odds with a number of practical considerations. Principally, in many surveillance settings optimal processing algorithms are either not known, or are computationally infeasible. Thus, improved performance often can be achieved with multi-stage processing that involves simpler and less computationally intensive algorithms than with centralized processing.

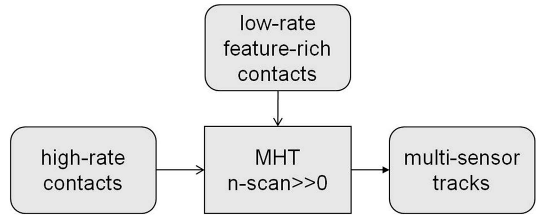


Fig. 1. *Track-while-fuse:* single-stage processing.

The multi-stage paradigm is seemingly at odds with fundamental results in the nonlinear filtering and distributed detection literature [13]. However, this is not actually the case. Rather, multi-stage approaches may outperform single-stage ones for two reasons: (1) like all trackers, single-stage tracking approaches are necessarily sub-optimal as they must contend with measurement origin uncertainty; and (2) measurement information is carried to downstream stages of multi-stage processing. As such, multi-stage processing as defined here is not in fact an instance of distributed processing.

A systems representation of *track-while-fuse*, *track-break-fuse*, and *track-before-fuse* is illustrated in Figs. 1–3. Note that use of these architectures need not require the availability of multi-sensor feeds: indeed our application of *track-break-fuse* (Section 4) is in the context of single-sensor surveillance.

In our two approaches to *multi-stage* MHT processing, we first seek to identify relevant, target-originated contact-level data from the high-rate sensor in a first tracker processing stage. We are not particularly concerned that multi-target interactions be handled properly. Indeed, the first-stage tracker need not be an MHT module, though it is important that it perform hard data association and that the module provide contact-level data associated with tracks.

In many multi-sensor settings, one has high-rate sensors (perhaps providing a scan every several seconds) that provide detection information but without much target feature information, if any. An example of this is a surveillance radar. Additionally, one may have a low-rate sensor (perhaps providing a scan every several hours) that provides detection information with significant target features, or attributes, besides kinematic information. An example of this is *synthetic aperture radar* (SAR) imagery that may provide target dimensions or target type. A standard single-stage processing architecture is illustrated in Fig. 1.

In the *track-break-fuse* architecture (Fig. 2), track labels are removed from the single-sensor tracks, and the resulting contact-level data are fed to the second-stage tracker along with the low-rate feature-rich contacts

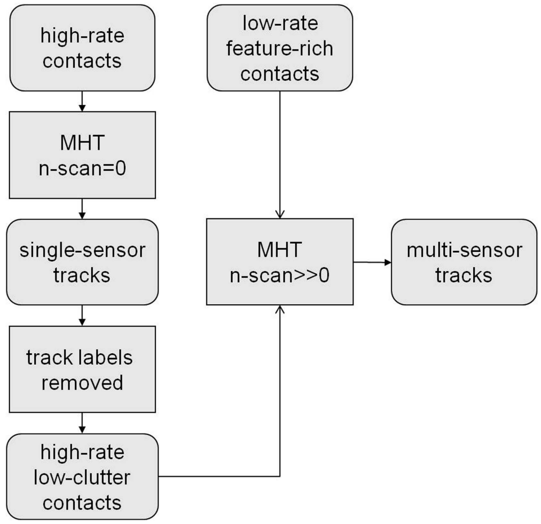


Fig. 2. *Track-break-fuse:* multi-stage processing with removal of track labels after first tracker module.

from the second sensor (if available). With the *track-before-fuse* architecture (Fig. 3), single-sensor tracks are fed to the second-stage MHT module along with the low-rate feature-rich contacts from the second sensor.

An important requirement for the *track-before-fuse* architecture is to have track-breakage logic in the second tracking module. Specifically, in instances in which the first-stage tracking has incorrectly introduced a track swap, subsequent feature information may identify that a tracking error has occurred. We wish to recover *gracefully* from the error, without the time-rollback solution that is operationally infeasible in large-scale surveillance. Error identification is prevalent when the feature-rich sensor has unambiguous target identification information. Correspondingly, when an infeasible update to a fused track is received, the fused track is terminated, and each upstream track of which the fused track is composed is subsequently treated as a new input track and made available for fusion with other fused tracks or for fused track initiation.

4. DENSE TARGET SCENARIOS AND THE TRACK-BREAK-FUSE ARCHITECTURE

A fundamental difficulty in target tracking is multi-target ambiguity, exhibited for example in a slowly-crossing target scenario. We find that MHT processing with large hypothesis tree depths improves tracking performance including a reduction of track swap occurrences. However, this poses a significant processing challenge as deep hypothesis trees are required. Indeed, hypothesis trees must be deep enough and rich enough so that the (local) track hypotheses associated with crossing and non-crossing tracks are maintained until ambiguities are resolved.

We will see that two-stage MHT processing with a *track-break-track* architecture does not impact tracking performance, it provides a dramatic computational benefit.

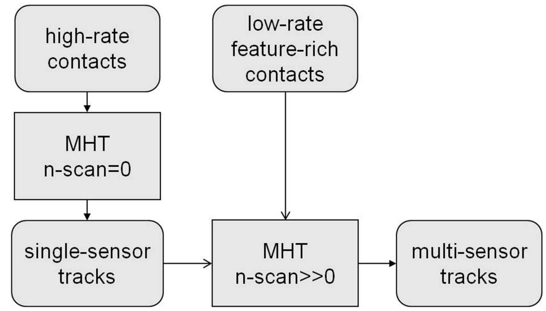


Fig. 3. *Track-before-fuse:* multi-stage processing with logical track breakage as needed in second tracker module.

The *track-break-track* architecture includes a first stage of tracking, followed by removal of all track label information and a second stage of MHT tracking applied only to those contacts that are included in the first-stage tracks. Thus, the first tracking stage can be regarded as a *filter* that identifies target-originated contacts. Multi-target association ambiguities are resolved in the second stage. The motivation for this architecture is that the first stage of processing can be executed quite effectively with no or small hypothesis tree depth, while the second stage requiring a larger hypothesis tree depth contends with much less contact data. Thus, we expect and find comparable performance to single-stage MHT in crossing-target scenarios, but at significant computation savings.

We now study the *percentage of success* for the crossing-target scenario, with *track-while-fuse* and *track-break-fuse* architectures and a range of hypothesis tree depths (*n-scan*). A key issue in this study is how we define *success* in a way that captures successful tracking through the target-crossing event. For the scenario of interest, this is well-captured by requiring a *track hold* or *track* P_D that exceeds 75 + % (note that in the case of a track swap, tracks are classified as false.)

Key parameters in this simulation are the following:

- *Target:* angle of approach = 22 deg; speed = 76.5 m/s;
- *Sensor:* $P_D = 0.7$, FAR = 10/scan; positional measurement error standard deviation—1 m in both x and y; scan rate = 1 Hz; number of scans = 150;
- *Tracker:* process noise = 10^{-3} m^{2}s^{-3}; initiation rule: 4-of-4; termination rule: 4 misses; association gate = 99%;
- *Monte Carlo settings:* 500 realizations of sensor data are generated. For each, six *track-while-fuse* tracker executions are performed (with *n-scan* from 0 to 5), as well as six *track-break-track* executions (*n-scan* = 0 in first stage, *n-scan* from 0 to 5 in the second stage).

Fig. 4 illustrates execution timing results. As expected, for small *n*-scan values, centralized tracking is faster. For larger *n*-scan values, *track-break-track* is faster. Fig. 5 illustrates tracking performance. With both tracking architectures, we find that there is increased tracking performance with increasing hypothesis depth, at the cost of increased execution time. We see

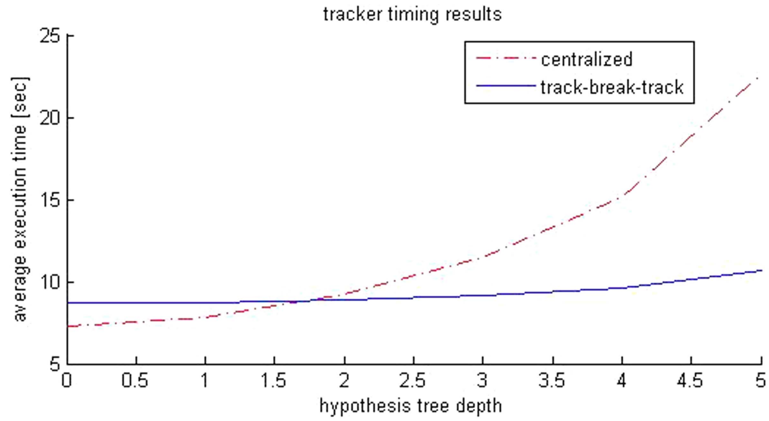


Fig. 4. Tracker timing results as a function of architecture and *n-scan* setting.

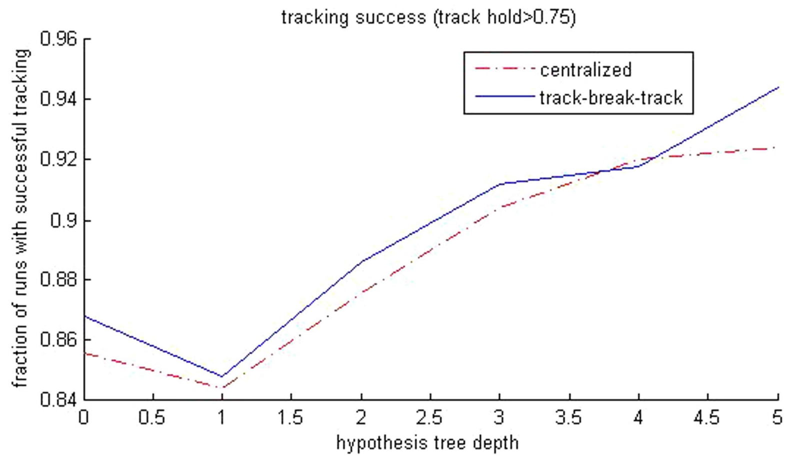


Fig. 5. Tracking performance as a function of architecture and *n-scan* setting.

that *track-break-fuse* matches the performance of *track-while-fuse* with significant computational savings.

An illustration of two tracker outputs for one run in the simulation study is illustrated in Figs. 6–7. Fig. 6 illustrates the sensor footprint with false contacts (black dots) and target-induced contacts (magenta dots). Target trajectories are in magenta, *track-while-fuse* tracks are in blue and red, indicating true and false tracks, respectively. (Note that the track swap occurrence is classified as a false track.) The (true) tracks resulting from *track-break-fuse* processing (*n-scan* = 5) are in green.

5. MULTI-SENSOR SURVEILLANCE AND THE TRACK-BEFORE-FUSE ARCHITECTURE

Fig. 8 illustrates the feature-aided tracking problem. We consider a situation where a high revisit rate sensor, e.g. surveillance radar, provides contact data to the fusion center. A second sensor provides contact data intermittently. Examples for the second sensor might include a SAR imaging sensor, a passive signal intelligence (SIGINT) sensor, or a passive transponder-based sensor such as the *automatic identification system* (AIS) in the maritime domain [12]. In the figure, target trajectories are shown in black. Surveillance radar contacts are shown in blue and black, for target detections and false returns, respectively. Intermittent, feature-rich returns are shown in red.

We consider again a target-crossing scenario. Sensor 1 has coverage of the entire surveillance region, while sensor two has coverage over a subset of the region that does not include the target-crossing event. The two sensors have the same nominal revisit rate, but sensor 2 is intermittent due to the more limited coverage. Both sensors provide positional measurements.

Sensor 2 is representative of a transponder-based passive sensor, like AIS. As such, it has a high revisit rate but intermittent coverage. While the detection probability is non-unity due to electromagnetic propagation effects and measurement collision with the time-division message allocation scheme [2], the false alarm rate is zero. Consistent with AIS data, sensor two provides precise target identification information.

Simulation parameters are given in Table I. Note that the track management criteria in the *track-while-fuse* and 1st stage of the multi-stage architectures differ slightly: this is done to achieve comparable data rates, and is required to account for intermittent coverage of sensor 2. The hypothesis tree depths in *track-while-fuse* and in the 2nd stage of the multi-stage architectures are chosen to be the same and sufficient to ensure good tracking

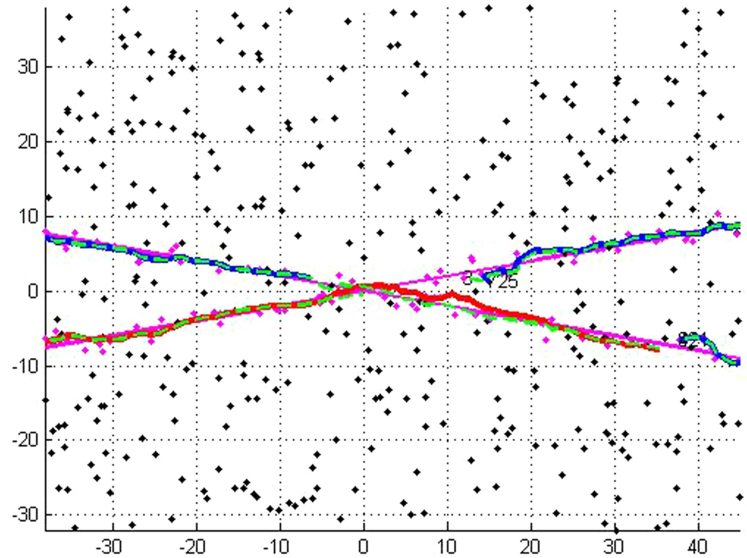


Fig. 6. One realization of sensor data, with two tracker outputs.

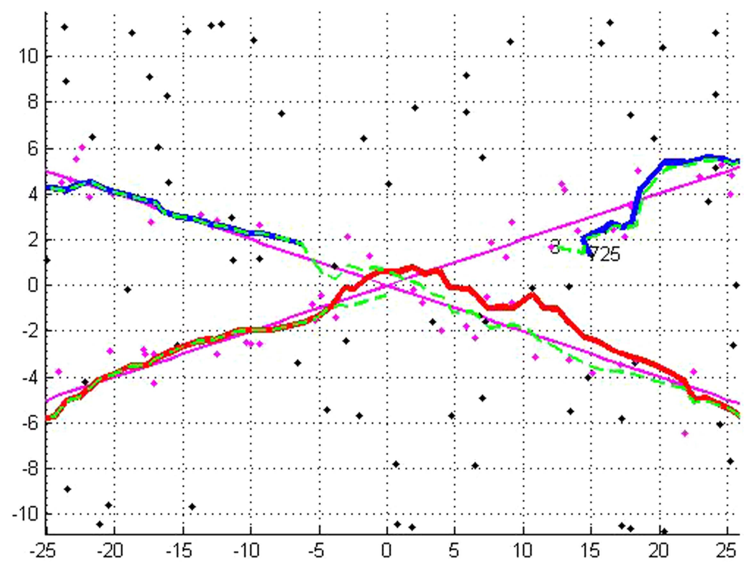


Fig. 7. Close-up view of *track-while-fuse* processing (red, blue) and *track-break-fuse* processing (green).

performance given the gap in sensor 2 coverage. The track break parameter is used to identify infeasible track updates that initiate fused track termination in the track-before-fuse architecture. The track classification metric is relevant to performance evaluation, as it is used to identify true tracks based on average localization error.

One scenario realization is illustrated in Figs. 9–12. In this instance, track swap has occurred in single-sensor (sensor 1) tracking. We see that single-stage *track-while-fuse* architecture does not exhibit the swap, nor does the multi-stage *track-break-fuse* architecture. The *track-before-fuse* architecture recovers from the upstream track swap by fragmenting the fused tracks, under the track-breakage logic described in Section 3.

As noted above, tracker performance evaluation relies on a track classification step whereby those tracks

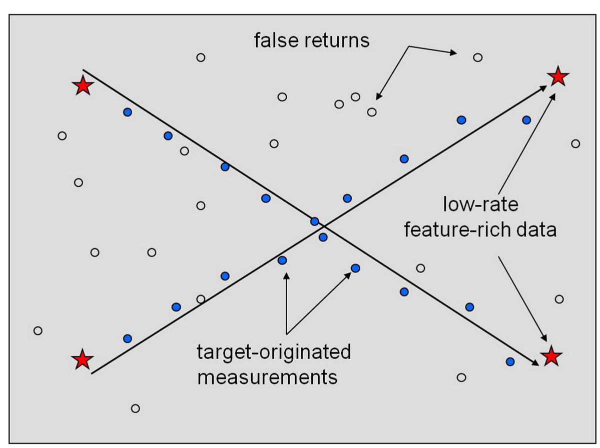


Fig. 8. A notional illustration of the feature-aided tracking problem.

TABLE I
Parameter Settings for the Simulation Study

Parameter	Setting
Monte Carlo realizations	200
Scenario duration	150 sec
Target number	2
Target start locations	$(-75\ \mathrm{m}, 5\ \mathrm{m})$, $(-75\ \mathrm{m}, -5\ \mathrm{m})$
Target velocities (until crossing)	$(1\ \mathrm{m/s}, -0.067\ \mathrm{m/s})$, $(1\ \mathrm{m/s}, 0.067\ \mathrm{m/s})$
Target velocities (after crossing)	$(1\ \mathrm{m/s}, -0.33\ \mathrm{m/s})$, $(1\ \mathrm{m/s}, 0.33\ \mathrm{m/s})$
Sensor 1 footprint	$(180\ \mathrm{m})^2$
Sensor 1 scan rate	1 Hz
Sensor 1 detection probability	0.8
Sensor 1 false alarm rate per scan	20
Sensor 1 measurement error covariance	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathrm{m}^2$
Sensor 2 footprint	$(180\ \mathrm{m})^2$ minus central swath, $\lvert x \rvert < 5$ m
Sensor 2 scan rate	1 Hz
Sensor 2 detection probability	0.8
Sensor 2 false alarm rate per scan	0
Sensor 2 measurement error covariance	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathrm{m}^2$
Track filter process noise parameter	$0.1\ \mathrm{m}^2\mathrm{s}^{-3}$
Track filter prior velocity covariance	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathrm{m}^2\mathrm{s}^{-2}$
Track correlation gate	99%
Track break (2nd stage track-before-fuse)	99.99%
Track initiation (track-while-fuse)	3-of-4
Track initiation (1st stage track-before-fuse & track-break-fuse)	4-of-4
Track kill (track-while-fuse)	6 misses
Track kill (1st stage track-before-fuse & track-break-fuse)	4 misses
Track kill (2nd stage track-before-fuse & track-break-fuse)	6 misses
N-scan (track-while-fuse)	10
N-scan (1st stage track-before-fuse & track-break-fuse)	0
N-scan (2nd stage track-before-fuse & track-break-fuse)	10
Track classification distance threshold	2 m

with sufficiently large average localization error with respect to all target trajectories are classified as false; otherwise, the closest target trajectory is identified. Subsequently, the following performance metrics are identified:

TABLE II
Performance Results

	Track-While-Fuse	Track-Before-Fuse	Track-Break-Fuse
PD	0.6067	0.9866	0.9861
FAR	23.40	22.07	42.35
FRAG	1.2229	2.1575	1.4175
ERROR	1.0348	1.061	0.9449
TIME	119.58	21.12	35.74

- *Track hold* (PD): ratio of total true track duration and total trajectory duration;
- *False track rate* (FAR): average number of false tracks [hr^{-1}];
- *Track fragmentation* (FRAG): average number of true tracks per tracked target;
- *Track localization error* (ERROR): average positional error between a true track and the corresponding target trajectory [m];
- *Tracker execution time* (TIME): average tracker execution time on a DELL OPTIPLEX GX620 with Intel Pentium D processor [sec]; note that 150 sec corresponds to real-time processing.

Performance results for the three feature-aided tracking architectures of interest are in Table II. An assessment of these results leads to the following conclusion:

- In terms of *track detection performance* (PD, FAR), *track-before-fuse* outperforms both *track-while-fuse* and *track-break-fuse*;
- In terms of *track continuity* (FRAG), the finding is reversed: *track-while-fuse* and *track-break-fuse* outperform *track-before-fuse*;
- In terms of *track accuracy* (ERROR), the architectures are comparable;
- In terms of *track computational load* (TIME), both multi-stage architectures perform significantly better than *track-while-fuse*.

Overall, we find that the *track-while-fuse* architecture has good performance, but it is not scalable to large hypothesis tree depths as would be required if the feature-rich sensor has a very low revisit rate, or has intermittent coverage with significant special gaps.

From a target-detection perspective, *track-break-fuse* performs comparably to *track-while-fuse*. In the simulation study, both track PD and track FAR are higher in *track-break-fuse*, since the effective track-level receiver operating characteristics (ROC) curve operating point is different between the two architectures. Similarly, the track fragmentation rate (FRAG) is roughly comparable. We conclude that the two architectures yield comparable tracking performance, but the *track-break-fuse* architecture exhibits significant computational savings.

The *track-before-fuse* architecture exhibits better target-detection performance than *track-break-fuse*. This

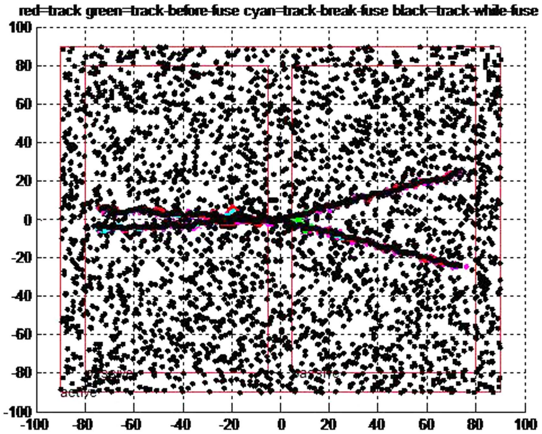


Fig. 9. One scenario realization.

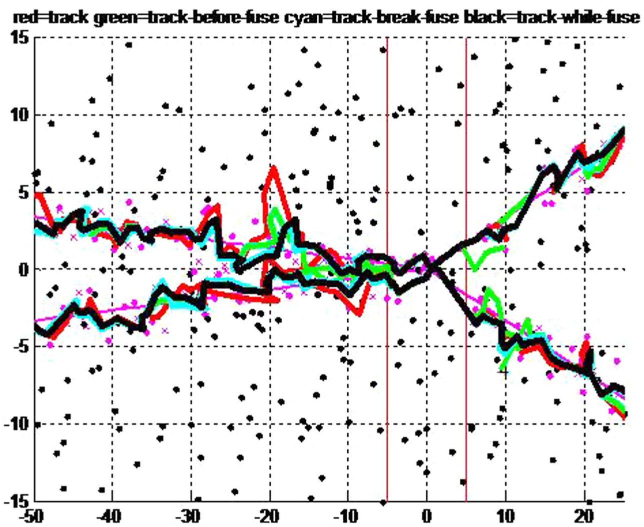


Fig. 10. Same realization as Fig. 9, with close-up on target crossing. Single-sensor (red) tracks exhibit swapping; track-before-fuse (green) exhibits track fragmentation after targets enter region of sensor 2 (feature-rich) coverage; both *track-break-fuse* (cyan) and *track-while-fuse* (black) are successful.

can be explained as follows: unlike track-break-fuse, there is no need to reacquire track in the second processing stage, since track associations are preserved. On the other hand, its track fragmentation rate is worse than that achieved with *track-break-fuse*, since first-stage association errors, when detected, lead to track termination and correspondingly to an increase in the overall fragmentation rate. As expected, the track-before-fuse architecture is the most efficient from a computational perspective.

To conclude, *track-while-fuse* is not scalable to large scenarios and large hypothesis-tree depths. Two feasible alternatives are *track-break-fuse* and *track-before-fuse*. Depending on the application, one or the other of these may be best.

6. CONCLUSIONS AND FUTURE DIRECTIONS

This paper proposes two multi-stage architectures for challenging surveillance problems that include dense target scenarios and multi-scale or intermittent multi-sensor coverage. In single-sensor benchmarking, the *track-break-fuse* architecture provides the same performance as track-while-fuse but with significant computational savings. In multi-sensor benchmarking, both multi-stage architectures achieve comparable track-level detection, localization, and track-continuity performance as single-stage, *track-while-fuse* processing. However, both architectures do so with dramatically reduced execution times. Further, the multi-stage architectures are extensible to larger hypothesis tree depths, while the single-stage architecture is not. The first multi-stage architecture, *track-before-fuse*, has higher track fragmentation than single-stage processing. The second multi-stage architecture, *track-break-fuse*, achieves comparable track fragmentation as in single-stage processing, at the cost of a small computational increase over *track-before-fuse*.

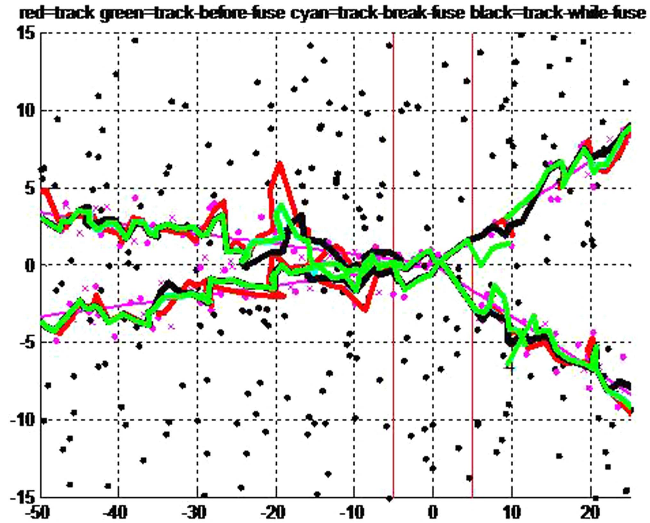


Fig. 11. Same as Fig. 9, with *track-before-fuse* result on top overlay (note fragmentation).

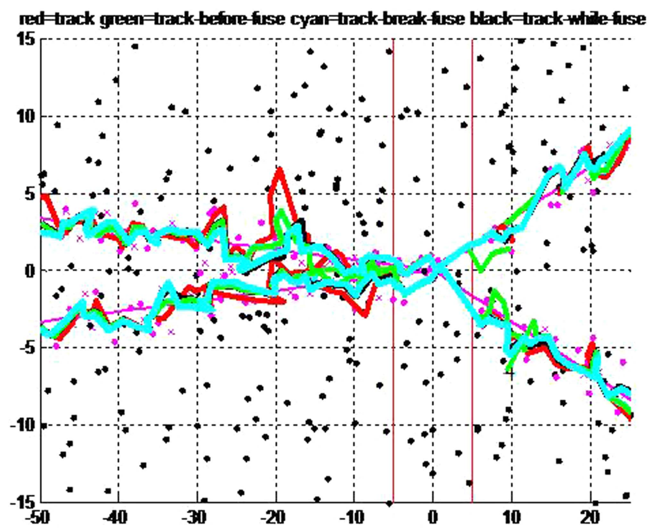


Fig. 12. Same realization as Fig. 9, with *track-break-fuse* result on top overlay (similar to *track-while-fuse*, with no fragmentation).

Future work will include analysis of an architecture that includes *some* track breakage after first-stage tracking, but less than the complete breakage prescribed under the track-break-fuse architecture. As such, this *hybrid* architecture should trade off the benefits of the *track-break-fuse* architecture (limited sensitivity to first-stage tracking errors) with those of the *track-before-fuse* architecture (computational savings, particularly in scenarios where multi-target association ambiguities persist for a long time).

Such a (hybrid) architecture would provide improved surveillance performance and would be particularly applicable to large sensor surveillance networks, where ambiguities may persist for a very large number of sensor scans, thus providing a flexible architecture for large-scale surveillance. The approach shares similarities with [4] but without requiring group-tracking logic.

REFERENCES

[1] S. Blackman and R. Popoli
Design and Analysis of Modern Tracking Systems.
Artech House, 1999.

[2] T. Calmettes, R. Challamel, O. Autran, and B. Garnier
Space-based AIS: Solutions to the simultaneous access issue.
In *Proceedings of the 3rd Annual Maritime Systems and Technology Conference*, Cadiz, Spain, Nov. 2008.

[3] C. Carthel, S. Coraluppi, K. Bryan, and G. Arcieri
Wide-area feature-aided tracking with intermittent multi-sensor data.
In *Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets*, Orlando FL, Apr. 2010.

[4] C-Y. Chong, et al.
Efficient multiple hypothesis tracking by track segment graph.
In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, WA, July 2009.

[5] S. Coraluppi and C. Carthel
Recursive track fusion for multi-sensor surveillance.
Information Fusion, **5**, 1 (Mar. 2004).

[6] S. Coraluppi, C. Carthel, P. Willett, M. Dingboe, O. O'Neill, and T. Luginbuhl
The track repulsion effect in automatic tracking.
In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, WA, July 2009.

[7] S. Gadaleta, A. Poore, S. Roberts, and B. Slocumb
Multiple hypothesis clustering and multiple frame assignment tracking.
In *Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets*, San Diego, CA, Aug. 2004.

[8] P. Horridge and S. Maskell
A scalable method of tracking targets with dependent distributions.
In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, WA, July 2009.

[9] T. Kurien
Issues in the design of practical multi-target tracking algorithms.
In Y. Bar-Shalom (Ed.), *Multi-Target Multi-Sensor Tracking: Advanced Applications*, Artech House, 1990.

[10] R. Popp, K. Pattipati, and Y. Bar-Shalom
M-best S-D assignment algorithm with application to multitarget tracking.
IEEE Transactions on Aerospace and Electronic Systems, **37**, 1 (Jan. 2001).

[11] D. Reid
An algorithm for tracking multiple targets.
IEEE Transactions on Aerospace and Electronic Systems, **24**, 4 (Dec. 1979).

[12] B. Tetreault
Use of the automatic identification system (AIS) for maritime domain awareness.
In *Proceedings of OCEANS 2005*, Washington, D.C., Sept. 2005.

[13] J. Tsitsiklis
Decentralized detection.
In *Advances in Signal Processing*, vol. 2, JAI Press, 1993.

Stefano Coraluppi received the B.S. degree in electrical engineering and mathematics from Carnegie Mellon University, Pittsburgh, PA in 1990, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, MD in 1992 and 1997.

He has held research staff positions at ALPHATECH Inc. in Burlington, MA (1997–2002), the NATO Undersea Research Centre in La Spezia, Italy (2002–2010), and Compunetix Inc. in Monroeville, PA (since August 2010). He has contributed to numerous programs in ground, undersea, and maritime surveillance for security and defense applications. His research interests include detection and estimation theory, multi-target tracking, data fusion, and stochastic control.

Dr. Coraluppi served as general cochair for the 9th ISIF/IEEE International Conference on Information Fusion (FUSION) in Florence, Italy in 2006 (with Peter Willett) and program chair for the NATO Workshop on Data Fusion and Anomaly Detection for Maritime Situational Awareness in La Spezia, Italy in 2009. He has served as member of the NATO Research and Technology Organization (RTO) Sensors and Electronics Technology (SET) Panel, associate editor for the *IEEE Journal of Oceanic Engineering* and the *ISIF Journal of Advances in Information Fusion (JAIF)*, and Lecturer for the NATO RTO, the Italian Naval Academy, and the University of Pisa. He was the 2010 ISIF President.

Currently, Dr. Coraluppi is associate editor for Target Tracking and Multisensor Systems for the *IEEE Transactions on Aerospace and Electronic Systems*, area editor for Tracking for the ISIF *JAIF*, Member of the Board of Directors of ISIF, and ISIF Vice President of Working Groups. He is a member of the ISIF Multistatic Tracking Working Group and the IEEE AESS Target Tracking Systems Panel. He is a Senior Member of the IEEE.

Craig Carthel received B.S. degrees in physics and mathematics in 1988, an M.S. in mathematics in 1992, and a Ph.D. in mathematics in 1995, all from the University of Houston, where he did research in numerical analysis and optimization theory.

He is a principal scientist at Compunetix Inc. in Monroeville, PA. From 1995 to 1997, he worked at the Institute for Industrial Mathematics at Johannes Kepler University, in Linz, Austria on parameter identification and inverse problems. From 1998 to 2002, he was a senior mathematician at ALPHATECH Inc. in Burlington, MA, where he worked on image processing, multisensor data fusion and ground target tracking. From 2002 to 2010, he was a senior scientist in the Applied Research Department at the NATO Undersea Research Centre in La Spezia, Italy, where he worked on military operations research, simulation, optimization, and data fusion problems associated with maritime environments. In 2006, he served as the technical program chair for the 9th International Conference on Information Fusion.

JAIF

Journal of Advances in Information Fusion

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

Volumes 1–5 Index

AUTHOR INDEX

SUBJECT INDEX

T

U

V

W

INTERNATIONAL SOCIETY OF INFORMATION FUSION

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

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 of papers in one of the technical areas listed on the inside cover of JAIF are encouraged to submit their papers for peer review at http://jaif.msubmit.net.

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 6 Number 1 June 2011 ISSN 1557-6418