June 2012 Volume 7 Number 1 ISSN 1557-6418

JAIF 12

Journal of Advances in Information Fusion

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

www.isif.org

ISIF

INTERNATIONAL SOCIETY OF INFORMATION FUSION

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

JOURNAL OF ADVANCES IN INFORMATION FUSION: JUNE 2012

EDITORS FOR TECHNICAL AREAS

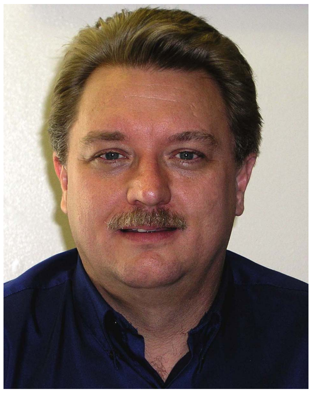
Manuscripts are submitted at http://jaif.msubmit.net. If in doubt about the proper editorial area of a contribution, submit it under the unknown area.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

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 2012



Where Are the Performance Guarantees for Information Fusion Algorithms?

In my editorial for the December 2011 issue of JAIF, I connected the low rate of transitions of information fusion techniques to real-world systems to the lack design methods for information fusion algorithms. Since engineers, not researchers, design and build systems, tools and design processes for design engineers are critical to the implementation of information fusion methods in real-world systems. Also, in that editorial, the development of effective design methods for information fusion algorithms was called out as the next frontier for the information fusion community. The focus of this editorial is on the need for performance guarantees for information fusion algorithms.

When one shops for an automobile, a long list of specifications and performance data are made available to the consumer. One of the most popular specifications for automobiles is the gas mileage and it is usually given in terms of miles per gallon for driving in the city and highway. While the miles per gallon varies with fuel quality, driving habits, etc., most consumers factor gas mileage specification into their purchasing decision. Another popular specification is the time required for the automobile to change speed from 0 to 60 miles per hour. Most all items in the market place have performance data and specifications provided for consumers to make purchasing decisions. On the other hand, no performance guarantee of any type is made available with information fusion algorithms. It is amazing that any of these algorithms get implemented in real systems. It is typically the reputations of the engineers and their professional opinions that lead to the implementation of new algorithms in a real system.

If you are still contemplating the reasons for the low rate of transitions of new algorithms to real systems, consider the last time that you purchased something of monetary significance with little or no insight into its value to you. Furthermore, consider the challenges of systems engineering when you have no performance characteristics for this important piece of your system called information fusion. How does one conduct a cost benefits analysis without any insight into the benefits?

Without a cost benefits analysis that supports the implementation of a new algorithm, the conventional algorithm will do just fine. Hence, new advances in information fusion will remain dormant.

Consider the well-studied and rather straightforward problem of tracking maneuvering targets. Prior to [1,2], the wealth of literature on this problem did not address achievement of a specified level of filter performance. In [1,2], the performance specifications were discussed with respect to setting an upper limit on the maximum mean squared error (MMSE) relative to the measurement error and minimizing the MMSE. The very basic tracking problem of linear measurements of a single maneuvering target in a scalar coordinate is addressed in [1,2]. Most tracking problems are significantly more complex. The methods were applied to radar tracking in [3] with some success. However, [1-3] addresses only a few of the issues associated with tracking performance. For tracking maneuvering targets with a radar or other nonlinear sensor, restrictions on the characteristics of the maneuver, the sensor-target geometry, and sensing requirements are needed be define the bounds for which the performance measures can be achieved. The tracking algorithm that can provide those performance measures with the fewest restrictions could be considered the best algorithm. However, for the most part, the performance specifications for tracking maneuvering targets is still an open problem.

The Tracker Operating Characteristic (TOC) and the average track life [4] are two metrics that have been proposed for characterizing the performance of tracking targets in the presence of false alarms. The TOC gives the average tracking accuracy of the Probabilistic Data Association Filter (PDAF) as a function of the probability of detection and false alarm density. Thus, given a false alarm density and probability of detection, the TOC gives the average tracking accuracy for the PDAF. Given a false alarm density and probability of detection, the average track life corresponds to the average number of measurements that will be included in a track before it is declared lost. A shortcoming of the TOC and average track life in [4] is the need to select the process noise standard deviation that was known to be an open issue during its publication.

Development of methods for specifying the performance of information fusion algorithms and algorithms that can guarantee that performance are important research problems for the information fusion community. The tracking of maneuvering targets and tracking of a single target in the presence of false alarms are the simplest of problems. MTT with finite sensor resolutions is extremely complex in comparison and this is the tracking problem of most real-world sensors. The development of design methods and performance characterization measures for MTT algorithms and other information fusion algorithms are difficult challenges that could be beyond the reach of our abilities, but the lack of these are clearly standing in the path toward the implementation of advanced algorithms in real-world systems.

William Dale Blair
Editor in Chief

[1] W. D Blair
Design of nearly constant velocity filters for tracking maneuvering targets.
In *Proceedings of 11th International Conference on Information Fusion*, Cologne, Germany, June 30–July 3, 2008.

[2] W. D Blair
Design of nearly constant velocity track filters for brief maneuvers.
In *Proceedings of 14th International Conference on Information Fusion*, Chicago, IL, July 5–8, 2011.

[3] W. D. Blair
Design of nearly constant velocity filters for radar tracking of maneuvering targets.
In *Proceedings of 2012 IEEE Radar Conference*, Atlanta, GA, May 7–11, 2012.

[4] Y. Bar-Shalom, P. K. Willett, and X. Tian
Tracking and data fusion: A handbook of algorithms.
YBS Publishing, Box U-4157, Storrs, CT, 06269-4157, 2011.

[5] W. D. Blair and Y. Bar-Shalom
Tracking maneuvering targets with multiple sensors: Does more data always mean better estimates?
IEEE Transactions on Aerospace and Electronic Systems, Jan. 1996, pp. 450–456.

Track-to-Track Association with Augmented State

RICHARD W. OSBORNE, III
YAAKOV BAR-SHALOM
PETER WILLETT

Association of tracks formed at different sensors is an ongoing area of interest in the field of information fusion and target tracking. In order to leverage additional information about a current target of interest that has been tracked at (an) additional sensor(s), track-to-track association (T2TA) must be performed. In addition to accurately identifying tracks with common origin, a desirable T2TA scheme will associate the tracks quickly, i.e., after only a few samples. A T2TA scheme is developed here that will take advantage of traditional kinematic state information as well as additional state information in the form of state augmentation. The main contribution is the use of two nonlinearly related state augmentations at the two sensors and accounting for their uncertainties. The results of T2TA are compared when using only kinematic state information, only state augmentation information, and the full augmented state. The full augmented state is shown to provide the most desirable association results, both in terms of accuracy and the number of samples needed to provide that accuracy.

Manuscript received May 13, 2011; revised September 7, 2011; released for publication October 19, 2011.

Refereeing of this contribution was handled by Stefano Coraluppi.

Authors' address: R. Osborne, III, Y. Bar-Shalom, and P. Willett, Department of Electrical and Computer Engineering, University of Connecticut, U-2157 Storrs, CT.

1. INTRODUCTION

Proper application of data association is a key step in multitarget-multisensor tracking. For targets that travel over large distances, a single sensor can rarely survey the entire trajectory. For those sensors that are tracking targets which have previously been tracked (or are still being tracked) at other sensors, the prior information available from the previous tracks can prove valuable. In order to utilize any of the available information from previously formed tracks, accurate track-to-track association (T2TA) must be performed.

Previous literature [4] has developed T2TA using the kinematic states of each track to form the likelihood that two tracks share a common origin. Additionally, [8] provides a general framework for performing T2TA by combining any additional information about the target that may be available.

Since the kinematic information of each track can be formed into a likelihood function, additional information about each track can also be incorporated into the association by forming another likelihood function from the additional information. We will call this extra information the "state augmentation," and, when combined with the kinematic state of the track, the full state will be known as the "augmented state."

This work will examine a method of forming a likelihood ratio based cost for T2TA using the augmented state and an evaluation of the accuracy of the resulting T2TA. The state augmentation available at each sensor need not be identical in nature, but there must be a probabilistic relationship between them, so that the likelihood of tracks sharing a common origin can be formed. The case of two sensors' state augmentations which are nonlinearly related will be the focus of this work, along with a method to account for their uncertainties. This work will also show that the full augmented state T2TA provides significantly better association (more accurate and earlier association) than the kinematic information can provide alone.

Section 2 will formulate the problem and define some notation. Sections 3.1 and 3.2 will review the likelihood ratio based cost for association with kinematic information, as well as develop the likelihood ratio based cost for the state augmentation information. Section 3.3 will further develop the state augmentation association cost by taking into account the uncertain estimates of the state augmentation. Section 3.4 will present the full augmented state association cost. Section 4 presents the target motion model and earth model used to estimate and predict the kinematic states of the targets. Section 5 describes the sensor measurement noise assumptions and necessary measurement conversions. Section 6 will present the simulation scenarios and result, and Section 7 will present the conclusions.

2. PROBLEM FORMULATION

A total of M sensors are assumed to track N targets, with trajectories which pass into and out of the surveillance regions of each sensor. Each sensor will individually form kinematic tracks on each target and provide additional information which will augment the kinematic state. Each target and sensor is assumed to follow a motion and measurement model of the form

$$x(k+1) = f(x(k)) + v(k) \tag{1}$$

$$z(k) = h(x(k)) + w(k) \tag{2}$$

where $x(k)$ is the kinematic state at time k, $z(k)$ is the measurement at time k, $v(k)$ is the process noise, and $w(k)$ is the measurement noise.

The full augmented state estimate of target i available from sensor m is

$$\hat{\mathbf{x}}_m^i(k) = \begin{bmatrix} \hat{x}_m^i(k) \\ \hat{y}_m^i(k) \end{bmatrix} \tag{3}$$

where $\hat{x}_m^i(k)$ is the estimate of the kinematic state of target i from sensor m. The quantity $\hat{y}_m^i(k)$ is the corresponding estimated state augmentation vector

$$y_m^i(k) = [\zeta_m^{i,1}(k), \zeta_m^{i,2}(k), \ldots, \zeta_m^{i,n}(k)]' \tag{4}$$

available at sensor m, consisting of n elements. The number of elements n need not be the same between sensors and may also be measures of different quantities entirely. The necessary information is the probabilistic relationship between the state augmentation at each sensor. For the case of $M = 2$, the association will be carried out by taking advantage of the pdf (probability density function) of the state augmentation at the second sensor conditioned on the augmentation at the first sensor and the target's kinematic state under the "common origin" assumption, i.e.,

$$p[y_2 \mid y_1, x^i] \tag{5}$$

For the purposes of this work, the kinematic state will be defined as a target's position and velocity components (in Cartesian coordinates). The second sensor's state augmentation will be the target's reflectivity.

The state augmentation likelihood ratio cost will be developed in Section 3.2 assuming $\hat{y}_m^i(k)$ is a known quantity, while Section 3.3 will present the case where $\hat{y}_m^i(k)$ is a vector of random variables.

In order to form the cost of associating two tracks from different sensors, the likelihood function of the tracks having common origin is needed. The likelihood function [3] of a parameter θ is defined as $p(z \mid \theta)$, where z denotes the observations. The maximum likelihood estimate of θ is then

$$\hat{\theta}^{ML}(z) = \arg\max_{\theta} p(z \mid \theta) \tag{6}$$

In the case of measurement-to-track association (M2TA) the observation will be the measurement [1], while for T2TA, the observation will be the kinematic state [2] and/or the state augmentation.

The cost of association requires a likelihood ratio in order to differentiate between common and disparate origin. Furthermore, in order to be a valid cost function, the cost of association should be the negative of the likelihood ratio. For these reasons, the cost will be given by the negative log-likelihood ratio (NLLR)

$$\text{NLLR} = -\ln\left\{\frac{p(z \mid \text{"common origin"})}{p(z \mid \text{"different origin"})}\right\} \triangleq \mathcal{L} \tag{7}$$

Minimizing the sum of the NLLRs for all the associations will provide the overall T2TA.

3. AUGMENTED STATE T2TA

The sequel will assume that T2TA takes place between two sensors ($M = 2$) to simplify the explanation of the costs, but the generalization to more than two sensors is straightforward. Each sensor will provide a list of tracks which are to be associated, and there exist a number of good methods to perform the assignment once the costs are obtained [6], [7], [12], [13]. For the two sensor case, a 2-D assignment algorithm such as the auction algorithm works quite well.

To perform the overall track association, the joint likelihood of two tracks having common origin is required. The joint likelihood of track j at the second sensor having originated from the same target as track i at the first sensor—assuming that the kinematic and state augmentation information are independent—is simply

$$\Lambda^{ij}(k) = \Lambda_x^{ij}(k)\Lambda_y^{ij}(k) \tag{8}$$

where $\Lambda_x^{ij}(k)$ is the kinematic likelihood and $\Lambda_y^{ij}(k)$ is the state augmentation likelihood.

3.1. Kinematic State T2TA

The kinematic likelihood of two tracks sharing a common origin has been derived previously [4], but will be briefly reviewed here. In order to determine if two tracks pertain to the same target using their kinematic information, the states can be directly compared. In some cases, the estimated state from one sensor may no longer be available (e.g., the target may be outside the first sensor's surveillance region). In that case, the predicted state may be used instead.

For two tracks which originated from the same target, the true states are equal, i.e.,

$$\Delta^{ij}(k) = x^i(k) - x^j(k) = 0 \tag{9}$$

The error of the state estimate difference is

$$\tilde{\Delta}^{ij}(k) = \tilde{x}^i(k) - \tilde{x}^j(k) \tag{10}$$

where $\tilde{x}^i(k)$ and $\tilde{x}^j(k)$ are the errors in the estimated states of track i and j, respectively. Assuming $\tilde{x}^i(k)$ and $\tilde{x}^j(k)$ are zero-mean, $\tilde{\Delta}^{ij}$ will also be zero-mean. The covariance of $\tilde{\Delta}^{ij}(k)$ is

$$T^{ij}(k) = E\{[\tilde{x}^i(k) - \tilde{x}^j(k)][\tilde{x}^i(k) - \tilde{x}^j(k)]'\} \tag{11}$$

$$= P^i(k) + P^j(k) - P^{ij}(k) - P^{ji}(k) \tag{12}$$

where P^i is the covariance of track i (at sensor 1), P^j is the covariance of track j (at sensor 2), and $P^{ij} = (P^{ji})'$ is the crosscovariance of tracks i and j [4]. Assuming the state estimation errors are Gaussian, the kinematic state likelihood function will be [2]

$$\Lambda_x^{ij}(k) = \frac{1}{|2\pi T^{ij}(k)|^{1/2}} \cdot \exp\left[-\frac{1}{2}(\hat{x}^i(k)-\hat{x}^j(k))'[T^{ij}(k)]^{-1}(\hat{x}^i(k)-\hat{x}^j(k))\right] \tag{13}$$

As stated previously, the cost used for the association is provided by a (dimensionless) likelihood ratio (specifically, the NLLR). This allows for the complete assignment of all the tracks from lists of different lengths, by allowing assignment of tracks to so called "dummy" tracks. The kinematic state likelihood function for assignment of a track to a nonexistent dummy track is

$$\Lambda_x^{0j}(k) = \lambda_{\mathrm{ex}} \tag{14}$$

where the dummy track is indexed by $i = 0$ and λ_{ex} is the spatial density of extraneous tracks seen by the second sensor (i.e., false or new tracks) [2].

The NLLR resulting from (13) and (14) is

$$\begin{aligned}\mathcal{L}_x^{ij}(k) &\triangleq -\ln\frac{\Lambda_x^{ij}(k)}{\Lambda_x^{0j}(k)}\\ &= \frac{1}{2}(\hat{x}^i(k)-\hat{x}^j(k))'[T^{ij}(k)]^{-1}(\hat{x}^i(k)-\hat{x}^j(k))\\ &\quad + \ln(\lambda_{\mathrm{ex}}|2\pi T^{ij}(k)|^{1/2})\end{aligned} \tag{15}$$

The overall cost of association using kinematic state information will explicitly handle the cases where dummy tracks are necessary, and is defined as

$$C_x^{ij}(k) = \begin{cases}\mathcal{L}_x^{ij}(k) & \text{if} \quad i,j \neq 0\\ -\ln(1-P_{D_1}) & \text{if} \quad i=0, \quad j\neq 0\\ -\ln(1-P_{D_2}) & \text{if} \quad i\neq 0, \quad j=0\end{cases} \tag{16}$$

where P_{D_m} is the track detection probability of the mth sensor.

3.2. State Augmentation T2TA

The likelihood of two tracks sharing a common origin using state augmentation information will now be derived. To illustrate the derivation, a specific example where the state augmentation $y_2^j(k)$ at the second sensor follows a Swerling 1 distribution will be used. In this case, the pdf of $y_2^j(k)$ is

$$p(y_2^j(k)\mid y_1^i) = \frac{1}{\bar{y}_2(g[y_1^i,\hat{x}^j(k)])}\exp\left\{-\frac{y_2^j(k)}{\bar{y}_2(g[y_1^i,\hat{x}^j(k)])}\right\} \tag{17}$$

where the (possibly nonlinear) function $\bar{y}_2(\cdot)$ is the predicted value of $y_2^j(k)$ and $g(\cdot)$ is a (possibly nonlinear) function of the first sensor's state augmentation.

The likelihood function of tracks i and j having common origin, using data over a window of length L_w, assuming $y_2^j(k)$ is independent across samples (due to the Swerling 1 assumption), is

$$\begin{aligned}\Lambda_y^{ij}(k) &= \prod_{l=k-L_w+1}^{k}\frac{1}{\bar{y}_2(g[y_1^i,\hat{x}^j(l)])}\cdot\exp\left\{-\frac{y_2^j(l)}{\bar{y}_2(g[y_1^i,\hat{x}^j(l)])}\right\}\\ &= \frac{1}{\prod_{l=k-L_w+1}^{k}\bar{y}_2(g[y_1^i,\hat{x}^j(l)])}\cdot\exp\left\{-\sum_{l=k-L_w+1}^{k}\frac{y_2^j(l)}{\bar{y}_2(g[y_1^i,\hat{x}^j(l)])}\right\}\end{aligned} \tag{18}$$

The likelihood function of (18) will provide the numerator of the likelihood ratio, and the likelihood of an assignment of a track at the second sensor to a dummy track will be the denominator. Such a likelihood will be given by

$$\begin{aligned}\Lambda_y^{0j}(k) &= \prod_{l=k-L_w+1}^{k}\frac{1}{\bar{y}_0^j(k)}\exp\left\{-\frac{y_2^j(l)}{\bar{y}_0^j(k)}\right\}\\ &= \frac{1}{(\bar{y}_0^j(k))^{L_w}}\exp\left\{-\sum_{l=k-L_w+1}^{k}\frac{y_2^j(l)}{\bar{y}_0^j(k)}\right\}\end{aligned} \tag{19}$$

where $\bar{y}_0^j(k)$ is the average of the observed y_2^j within a window of length L_w ending at k (i.e., the average of an "extraneous origin" [1] signal)

$$\bar{y}_0^j(k) = \frac{1}{L_w}\sum_{l=k-L_w+1}^{k}y_2^j(l) \tag{20}$$

The NLLR resulting from (18) and (19) is

$$\begin{aligned}\mathcal{L}_y^{ij}(k) &\triangleq -\ln\frac{\Lambda_y^{ij}(k)}{\Lambda_y^{0j}(k)}\\ &= -L_w\ln(\bar{y}_0^j(k)) - \sum_{l=k-L_w+1}^{k}\frac{y_2^j(l)}{\bar{y}_0^j(k)}\\ &\quad - \frac{y_2^j(l)}{\bar{y}_2(g[y_1^i,\hat{x}^j(l)])} - \ln[\bar{y}_2(g[y_1^i,\hat{x}^j(l)])]\end{aligned} \tag{21}$$

The overall cost of association using state augmentation information will, once again, explicitly handle the cases where dummy tracks are necessary, and is defined as

$$C_y^{ij}(k) = \begin{cases}\mathcal{L}_y^{ij}(k) & \text{if} \quad i,j \neq 0\\ -\ln(1-P_{D_1}) & \text{if} \quad i=0, \quad j\neq 0\\ -\ln(1-P_{D_2}) & \text{if} \quad i\neq 0, \quad j=0\end{cases} \tag{22}$$

where P_{D_m} is the track detection probability of the mth sensor.

3.3 Modified Cost Function for Uncertainty in the Estimated State Augmentation

The previous section derived the cost of association based on state augmentation information by implicitly assuming that the first sensor's state augmentation was a known quantity. In reality, the first sensor will only be able to estimate the state augmentation, and the overall association can be improved by accounting for this uncertainty. To account for the uncertainty, the expected value of (17) will be used in the calculation of the state augmentation cost. In general, the function $g(\cdot)$—which determines the expected value of the second sensor's state augmentation and is itself a function of the first sensor's *estimated* state augmentation $\hat{y}_1^i$—is nonlinear. The expectation of (17) over y_1^i will, therefore, be handled by an unscented transform [9] and is

$$\begin{aligned} E_{y_1^i}[p(y_2^j(k) \mid y_1^i)] &= \int p(y_2^j(k) \mid y_1^i) p(y_1^i) dy_1^i \\ &= \int p(y_2^j(k) \mid g(\hat{y}_1^i, \hat{x}^j(k))) p(y_1^i) dy_1^i \end{aligned} \tag{23}$$

where $g(\hat{y}_1^i, \hat{x}^j(k))$ is the mapping of the state augmentation from sensor 1 to the state augmentation of sensor 2 (i.e., the sufficient statistic for the predicted state augmentation at sensor 2, which is the target reflectivity) at the position component of $\hat{x}^j(k)$.

This is approximated by using a probability mass function

$$E_{y_1^i}[p(y_2^j(k) \mid y_1^i)] \approx \sum_{m=-n}^{n} p(y_2^j(k) \mid g(y_{1m}^i, \hat{x}^j(k))) \mu(y_{1m}^i) \tag{24}$$

where n is the dimension of y_1^i and $\mu(y_{1m}^i)$ are the weights (point masses) of the $2n+1$ points used in the unscented transform [9].

In the case of $n = 5$, the points and weights will be

$$\begin{aligned} y_{1m}^i &= \hat{y}_1^i + \mathrm{sgn}(m)\sqrt{7}\sigma_m \\ \mu(y_{1m}^i) &= \begin{cases} 1/14 & \text{if } m \neq 0 \\ 2/7 & \text{if } m = 0 \end{cases} \end{aligned} \tag{25}$$

where $\hat{y}_1^i$ is the state augmentation handed over from the first sensor, and σ_m is the standard deviation for the mth element of $\hat{y}_1^i$.

The weights for each point of the unscented transform according to [9] should be

$$\mu_m = \begin{cases} \kappa/(n+\kappa) & \text{if } m = 0 \\ 1/2(n+\kappa) & \text{if } m \neq 0 \end{cases} \tag{26}$$

where κ is an extra degree of freedom for "fine tuning" the approximation. In this case, $\kappa = 2$ was chosen, since, for a one dimensional Gaussian distribution, that particular choice of κ will provide an approximation which coincides to matching the second and fourth moments of the Gaussian distribution.[1]

Utilizing the unscented transform, the state augmentation cost function will be calculated as in Section 3.2, replacing (17) with (24).

3.4. Full Augmented State T2TA

In light of the previous derivations, the cost of the association using the full augmented state can now be fully specified. Assuming the kinematic and state augmentation information is independent, the full augmented state cost is [4]

$$C^{ij}(k) = \begin{cases} \mathcal{L}_x^{ij}(k) + \mathcal{L}_y^{ij}(k) & \text{if } i, j \neq 0 \\ -\ln(1 - P_{D_1}) & \text{if } i = 0, \quad j \neq 0 \\ -\ln(1 - P_{D_2}) & \text{if } i \neq 0, \quad j = 0 \end{cases} \tag{27}$$

Note that the costs in lines 2 and 3 of (27) are not the sums of the corresponding lines of (16) and (22), since, if a detection is missing, both point kinematic and state augmentation are missing.

With the NLLR costs fully specified, the two dimensional assignment of tracks from the first sensor to tracks at the second sensor—each list with a dummy track added—can now be carried out using the auction algorithm (for a review of the auction algorithm see [12]).

4. TARGET MOTION MODEL

In order to estimate and predict the target state for use in the kinematic cost function, an appropriate target motion model is necessary. All tracking performed here is assumed to be done on targets which are under the influence of gravity alone. Since gravity is the only force acting on the targets, the gravitational pull of the earth must be accurately modeled.

An ellipsoidal earth model with a single zonal harmonic term will be used to model the acceleration of a target due to the earth's gravity [5], [10]. Additionally, the tracking is assumed to be done in the earth-centered-inertial (ECI) coordinate system, which avoids the need to calculate accelerations due to Coriolis and centrifugal forces caused by the earth's rotation if an earth-fixed coordinate system were used. The acceleration due to gravity using this ellipsoidal earth model is

$$\begin{aligned} \mathbf{a}(\mathbf{p}) &\triangleq \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \\ &= -\frac{\mu}{p^3} \begin{bmatrix} \left(1 + \frac{3J_2 r_e^2}{2p^2}\left(1 - 5\frac{z^2}{p^2}\right)\right) x \\ \left(1 + \frac{3J_2 r_e^2}{2p^2}\left(1 - 5\frac{z^2}{p^2}\right)\right) y \\ \left(1 + \frac{3J_2 r_e^2}{2p^2}\left(3 - 5\frac{z^2}{p^2}\right)\right) z \end{bmatrix} \end{aligned} \tag{28}$$

[1] The authors of [9] suggest using $n + \kappa = 3$ for a Gaussian distribution, but this would give rise, for $n = 5$, to a nonsensical negative weight for the center point of the approximation.

where $\mu = 398601.2\ \text{km}^3/\text{s}^2$ is earth's gravitation constant, $J_2 = 1.0826 \times 10^{-3}$ is the first zonal harmonic term, r_e is the equatorial radius of the earth (using the WGS84 earth model, $r_e = 6378.137$ km), and $p = \|\mathbf{p}\|$ where $\mathbf{p} = [x, y, z]'$.

The target motion model which will be assumed for estimation and prediction will be a discretized continuous white noise acceleration model (DCWNA), with the acceleration due to earth's gravity assumed piecewise constant and entering the motion model as a known input. The acceleration at each sampling time will be given by (28) evaluated at the current measurement of the target's position (if a current measurement is not available, the predicted state will instead be used). Given the state $\mathbf{x}(k)$ at time k, where $\mathbf{x}(k) \overset{\Delta}{=} [x(k), y(k), z(k), \dot{x}(k), \dot{y}(k), \dot{z}(k)]'$ and is given in ECI coordinates, the state $\mathbf{x}(k+1)$ is

$$\mathbf{x}(k+1) = F\mathbf{x}(k) + G\mathbf{a}(\mathbf{p}(k)) + v(k) \tag{29}$$

where

$$F = \begin{bmatrix} I_3 & I_3 T \\ 0 & I_3 \end{bmatrix}, \qquad G = \begin{bmatrix} I_3 \dfrac{T^2}{2} \\ I_3 T \end{bmatrix} \tag{30}$$

I_3 is the 3×3 identity matrix, $\mathbf{p}(k)$ is the target position (given by measurement or predicted state) at time k and $v(k)$ is the process noise.

Since the targets are all assumed to be under the influence of gravity alone, a very small process noise should be sufficient. The covariance of the process noise $v(k)$ is given by

$$Q = \begin{bmatrix} I_3 \dfrac{T^3}{3} & I_3 \dfrac{T^2}{2} \\ I_3 \dfrac{T^2}{2} & I_3 T \end{bmatrix} q \tag{31}$$

The intensity (power spectral density) of the process noise q chosen for this motion model was $0.01\ \text{m}^2/\text{s}^3$.

The target tracking will be performed using a standard Kalman filter [3] with the above motion model. The prediction from the end of the first sensor's tracks to the time the second sensor begins tracking will be done using the same DCWNA model with the acceleration due to gravity given by (28).

5. SENSOR MEASUREMENT NOISE AND MEASUREMENT CONVERSION

While the target measurements used in the Kalman filter will be in ECI coordinates, the measurement noise will be added to the measurements in terms of range, azimuth and elevation. The measurement of the target is given by

$$r_m = r + w_r, \qquad a_m = a + w_a, \qquad \epsilon_m = \epsilon + w_\epsilon \tag{32}$$

where r, a and ϵ are the true range, azimuth and elevation angles, respectively, and w_r, w_a and w_ϵ are independent zero-mean Gaussian noise with standard deviations σ_r, σ_a and σ_ϵ, respectively.

TABLE I
Measurement Noise Standard Deviations for Both Sensors
(The Handover was from Sensor 1 to Sensor 2)

	Range σ_r (m)	Azimuth σ_α (mrad)	Elevation σ_ϵ (mrad)
Scenario 1	10	0.5	0.5
Scenario 2	20	1	1

TABLE II
State Augmentation Estimation Error Standard Deviations

	σ_{ζ_1}	σ_{ζ_2}	σ_{ζ_3}	σ_{ζ_4}	σ_{ζ_5}
Scenario A	17.5	17.5	17.5	0.087	17.5
Scenario B	52.5	52.5	52.5	0.087	52.5
Scenario C	52.5	52.5	52.5	0.140	52.5

These measurements must be converted to ECI coordinates in order to be used by the Kalman filter which uses the models outlined in Section 4. The measurements should first be converted to a Cartesian coordinate system aligned with the sensor face plane, and then rotated and translated to match the ECI coordinate system. Since the range from the sensor to the targets is very large, there is a possibly significant bias that can be introduced in the conversion from spherical coordinates to Cartesian. The unbiased conversion [11] will, therefore, be used.

The covariance matrix in ECI coordinates, R_e, will then be

$$R_e = T(k)' R_r T(k) \tag{33}$$

where $T(k)$ is the transformation matrix used to rotate from ECI coordinates to sensor face coordinates at time k and R_r is the covariance matrix of the unbiased converted measurements.

6. SIMULATION RESULTS

The scenarios considered consist of 6 targets under the influence of gravity alone. The two sensors are both assumed to measure reflectivity, but have different sampling intervals. The first sensor tracks the targets over approximately 520 samples (some of the targets have a few extra samples at the beginning or end of the track). The interval between the time the first sensor stops tracking and the second sensor begins tracking is approximately 1500 sampling intervals. The second sensor tracks the targets over approximately 1060 samples. The targets at all times are separated by more than $40\sigma_r$, i.e., they are resolved. The measurement noise standard deviations of the two sensors are given in Table I.

Additionally, the $n = 5$ elements of y_1^i estimated for each track at the first sensor are assumed to have zero-mean Gaussian errors with standard deviations given in Table II. The scenarios will be designated as 1A–1C and 2A–2C.

Each figure shows the average association accuracy (i.e., fraction of tracks correctly identified as having common origin), averaged over 200 runs. Three association accuracy plots are shown in each figure: the association accuracy using only the kinematic cost (16), the association accuracy using only the state augmentation cost (22), and the association accuracy using the full augmented state cost (27).

Figures 1–3 show the association accuracy for scenarios 1A–1C, respectively (with measurement noise standard deviations given by scenario 1 of Table I, and state augmentation error standard deviations given by scenarios A–C of Table II) with window length $L_w = 100$.

Figures 4–6 show the association accuracy for scenarios identical to figures 1–3, however, the unscented evaluation of the state augmentation costs have been used.

Figures 7–9 show the association accuracy for scenarios 1A–1C, respectively (with measurement noise standard deviations given by scenario 1 of Table I, and state augmentation error standard deviations given by scenarios A–C of Table II) with window length $L_w = 200$.

Figures 10–12 show the association accuracy for scenarios identical to figures 7–9, however, the unscented evaluation of the state augmentation costs have been used.

Figures 13–15 show the association accuracy for scenarios 2A–2C, respectively (with measurement noise standard deviations given by scenario 2 of Table I, and target state augmentation error standard deviations given by scenarios A–C of Table II) with window length $L_w = 100$.

Figures 16–18 show the association accuracy for scenarios identical to figures 13–15, however, the unscented evaluation of the state augmentation costs have been used.

Figures 19–21 show the association accuracy for scenarios 2A–2C, respectively (with measurement noise standard deviations given by scenario 2 of Table I, and target state augmentation error standard deviations given by scenarios A–C of Table II) with window length $L_w = 200$.

Figures 22–24 show the association accuracy for scenarios identical to figures 19–21, however, the unscented evaluation of the state augmentation costs have been used.

When the measurement noise is large, the accuracy of the kinematic tracking will suffer, and the kinematic-only association will remain accurate, but will take longer to achieve high accuracy. The accuracy of the state augmentation only association remains relatively unaffected by the highest measurement noise (scenario 2), but even higher noise could begin to affect the accuracy because of inaccurate target positions used in determining the aspect angles of the targets.

The higher levels of state augmentation estimation uncertainty can degrade the performance of the augmentation association enough to cause the combined association to perform worse than the association using kinematic information alone. The use of very inaccurate state augmentation variables clouds the picture by flattening the likelihoods, and deteriorates the association accuracy. A larger window size of $L_w = 200$ can improve the association somewhat and prevent some degradation of the combined accuracy for the higher levels of state augmentation estimation error, as seen in Figs. 7–12 and 19–24. Ultimately, the overall accuracy is very sensitive to the state augmentation estimation accuracy.

Figures 4–6, 10–12, 16–18, and 22–24, however, show that the unscented transform can improve the overall accuracy for the cases of higher state augmentation estimation errors by taking into account the uncertainty of the estimation.

The scenarios with higher measurement noisepresent some interesting results for the early portions of the second sensor's tracks. The association based on only kinematic information will completely fail to accurately associate any of the second sensor's targets for the early portion of the trajectories. The augmentation-only association will perform more accurately than kinematic-only association for those early portions, but will not perform as well as desired. Once the two costs, which work poorly alone, are combined, however, the association is extremely accurate. The reason for this is the kinematic information results in a cost function which favors assigning the second sensor's tracks to dummy tracks for the early portions of the trajectories. This results in the very poor association accuracy in the kinematic-only association. The augmentation-only association results in costs which favor the actual tracks over dummy tracks, but has difficulty correctly distinguishing among the true tracks. When the negative costs from the augmentation portion are added to the kinematic costs, however, the total costs no longer favor dummy tracks (due to the augmented state information) and are still distinguishable from early portions of the trajectories (due to the kinematic information).

Even when the measurement noise standard deviations are small (in which case the kinematic-only association performs very well), the addition of the state augmentation into the association improves the accuracy of the T2TA, resulting in a near perfect association very quickly after the targets are tracked at the second sensor. The use of the state augmentation cost also improves T2TA accuracy even though the augmentation-only association is less accurate than the kinematic-only association. When the measurement noise standard deviations are larger, the addition of the state augmentation cost helps to improve the overall T2TA, even during portions of the track when the kinematic-only association performs very poorly.

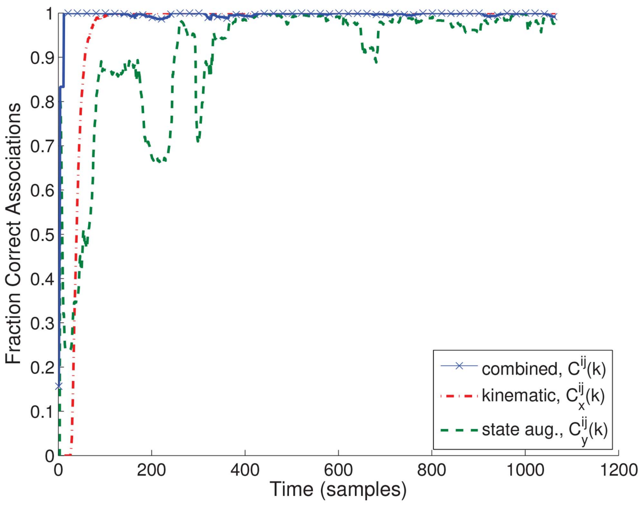


Fig. 1. Average association accuracy over 200 runs (scenario 1A), with window length $L_w = 100$.

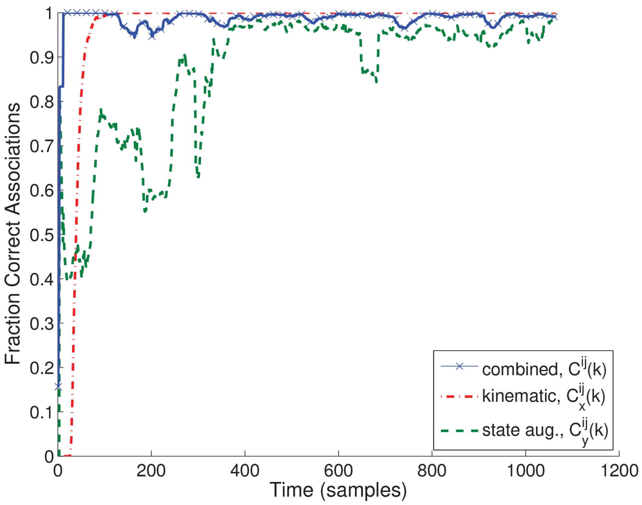


Fig. 2. Average association accuracy over 200 runs (scenario 1B), with window length $L_w = 100$.

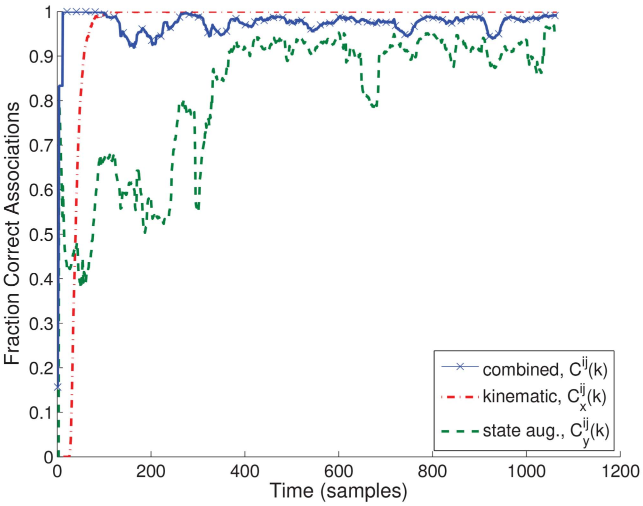


Fig. 3. Average association accuracy over 200 runs (scenario 1C), with window length $L_w = 100$.

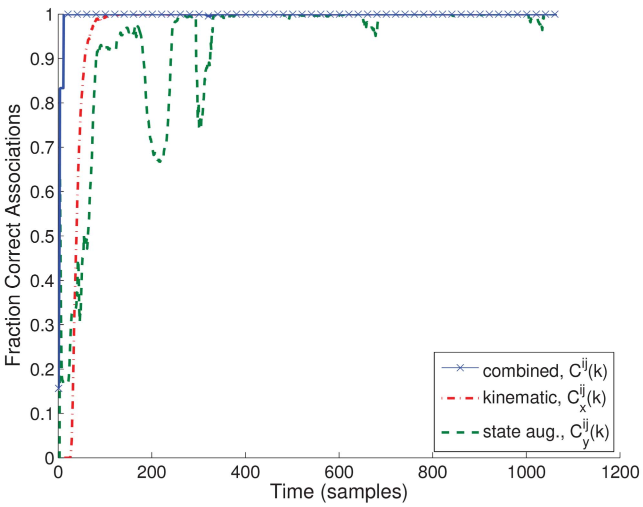


Fig. 4. Average association accuracy over 200 runs with unscented state augmentation costs (scenario 1A), window length $L_w = 100$.

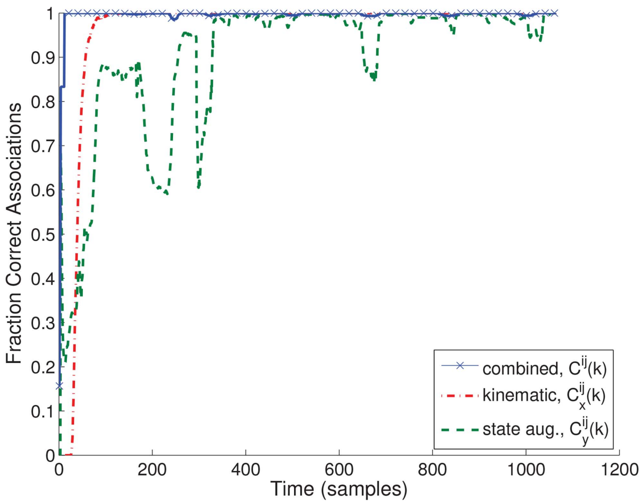


Fig. 5. Average association accuracy over 200 runs with unscented state augmentation costs (scenario 1B), window length $L_w = 100$.

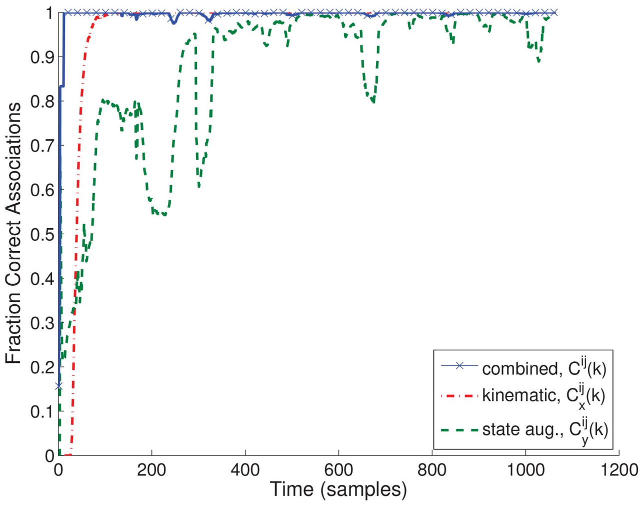


Fig. 6. Average association accuracy over 200 runs with unscented state augmentation costs (scenario 1C), window length $L_w = 100$.

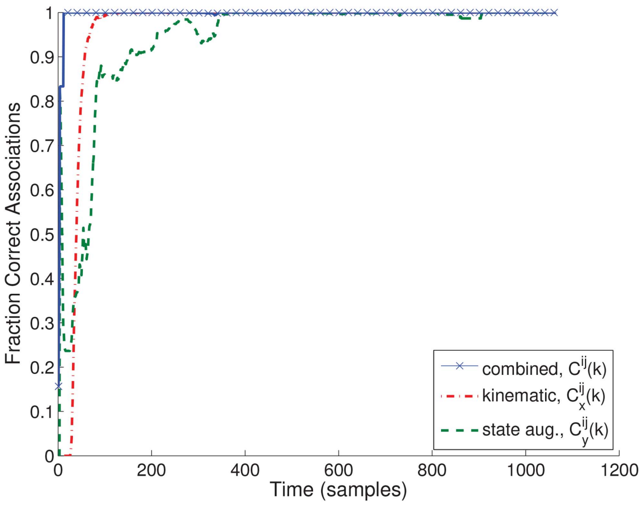


Fig. 7. Average association accuracy over 200 runs (scenario 1A), with window length $L_w = 200$.

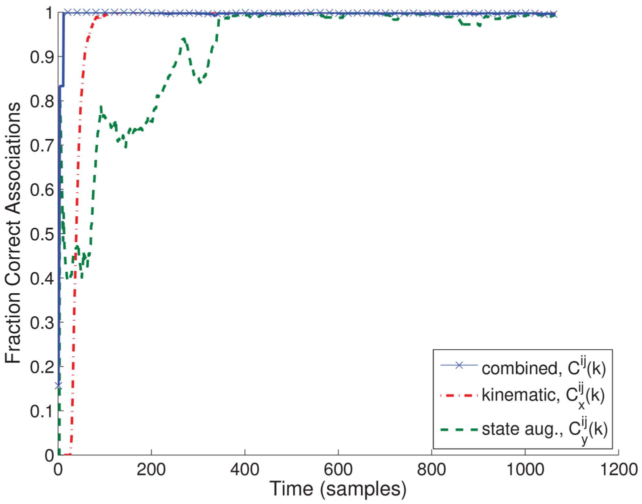


Fig. 8. Average association accuracy over 200 runs (scenario 1B), with window length $L_w = 200$.

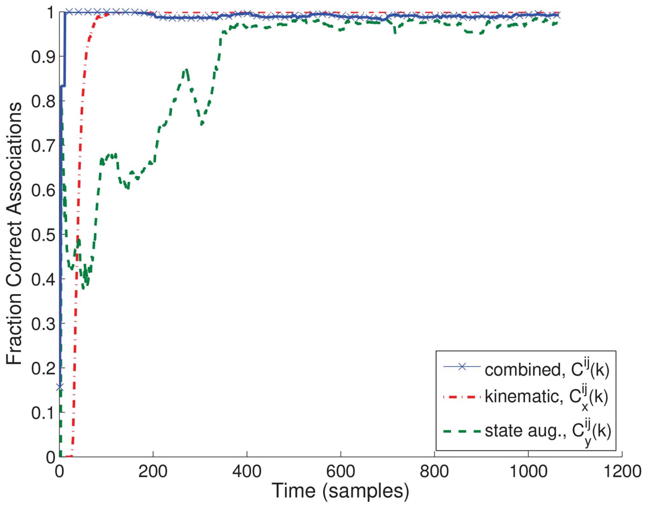


Fig. 9. Average association accuracy over 200 runs (scenario 1C), with window length $L_w = 200$.

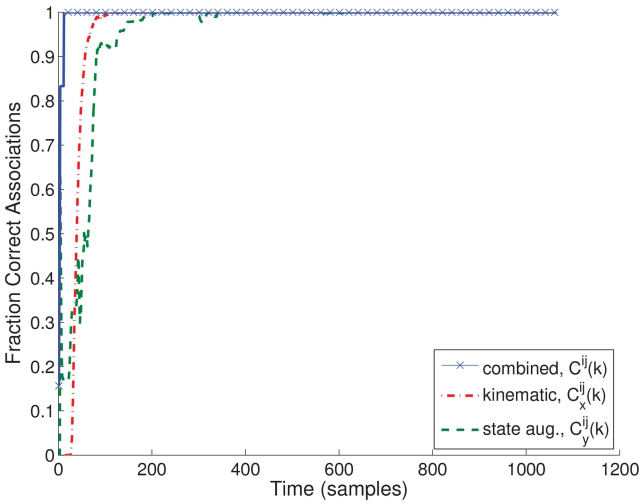


Fig. 10. Average association accuracy over 200 runs with unscented state augmentation costs (scenario 1A), window length $L_w = 200$.

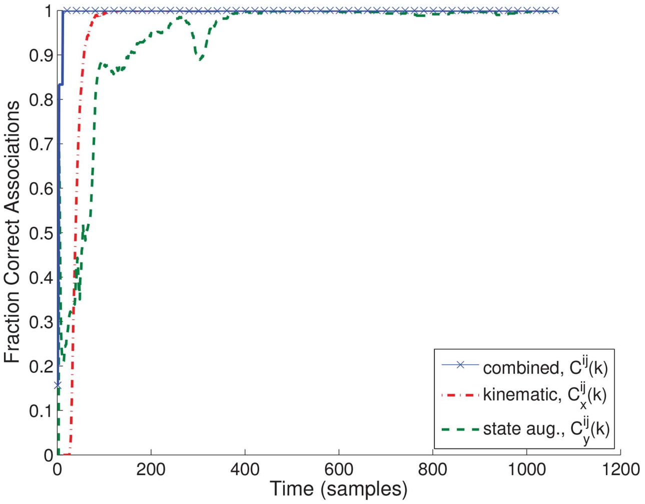


Fig. 11. Average association accuracy over 200 runs with unscented state augmentation costs (scenario 1B), window length $L_w = 200$.

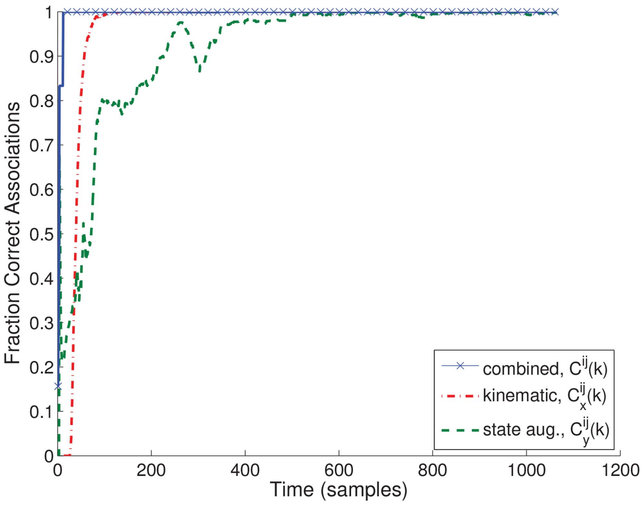


Fig. 12. Average association accuracy over 200 runs with unscented state augmentation costs (scenario 1C), window length $L_w = 200$.

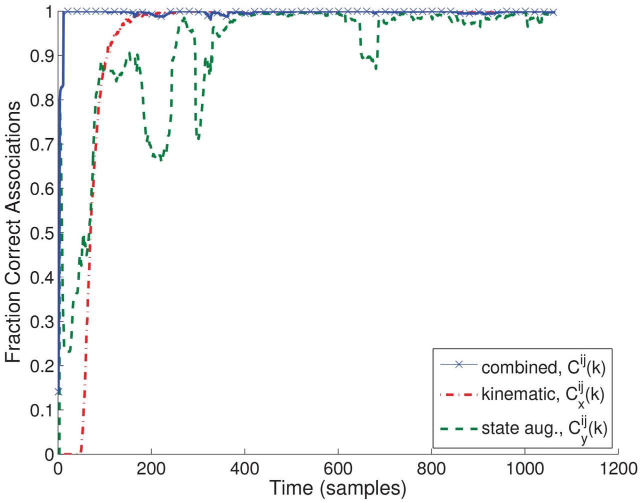


Fig. 13. Average association accuracy over 200 runs (scenario 2A), with window length $L_w = 100$.

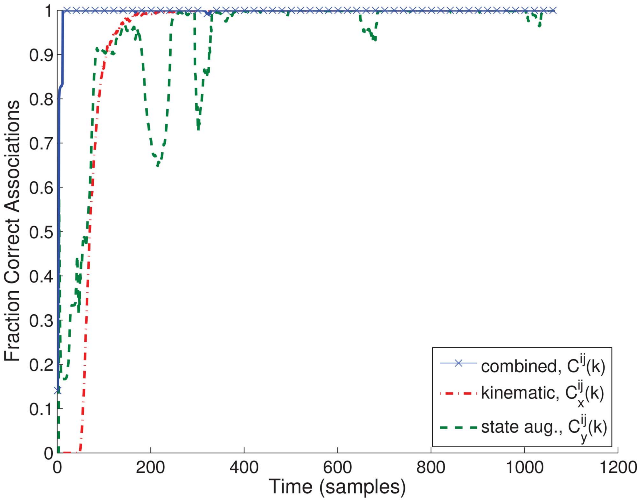


Fig. 16. Average association accuracy over 200 runs with unscented state augmentation costs (scenario 2A), window length $L_w = 100$.

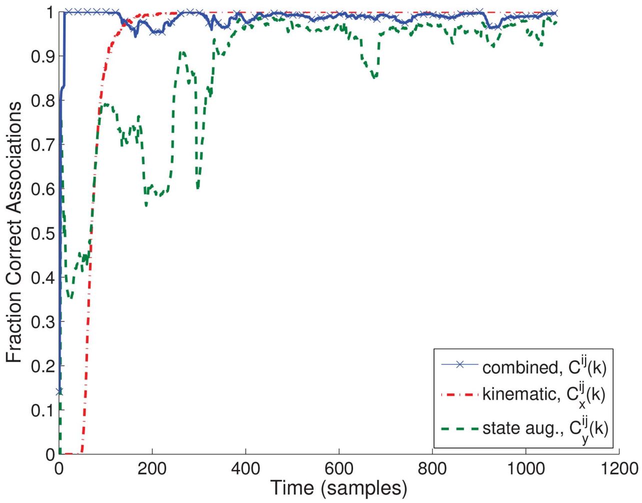


Fig. 14. Average association accuracy over 200 runs (scenario 2B), with window length $L_w = 100$.

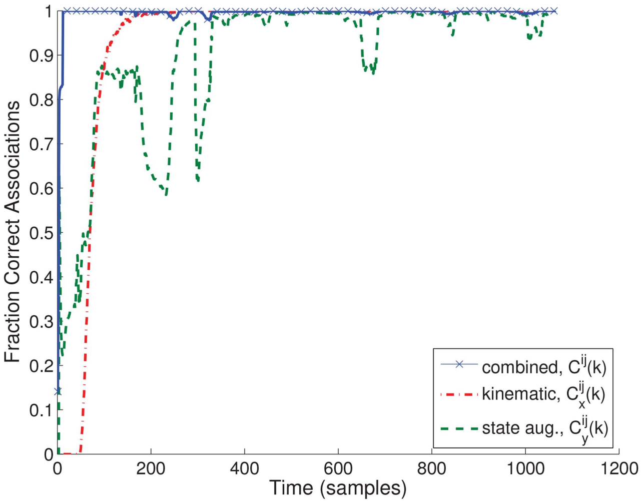


Fig. 17. Average association accuracy over 200 runs with unscented state augmentation costs (scenario 2B), window length $L_w = 100$.

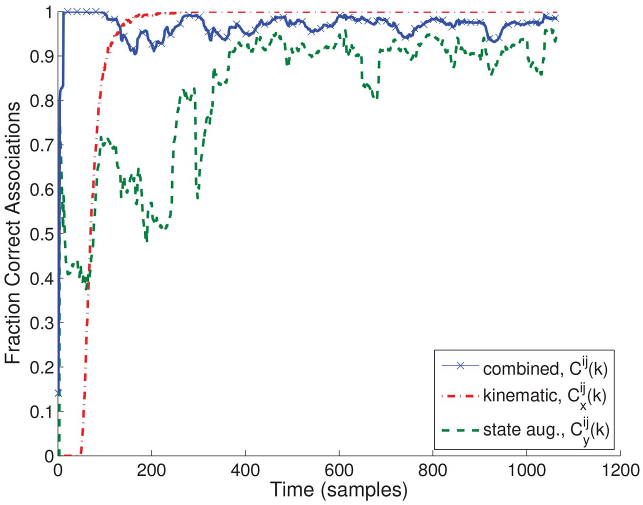


Fig. 15. Average association accuracy over 200 runs (scenario 2C), with window length $L_w = 100$.

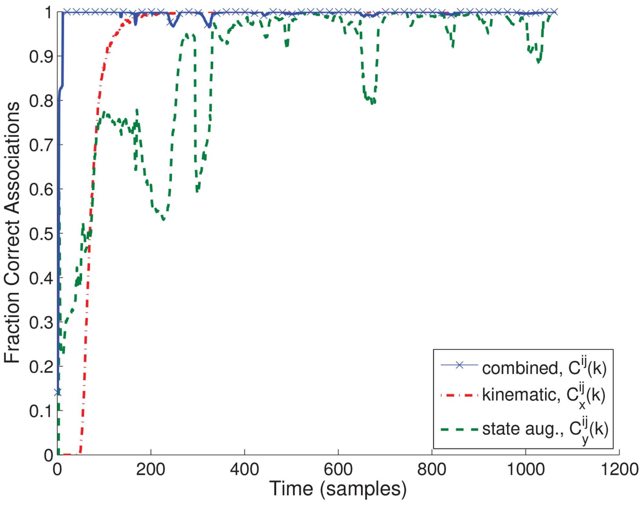


Fig. 18. Average association accuracy over 200 runs with unscented state augmentation costs (scenario 2C), window length $L_w = 100$.

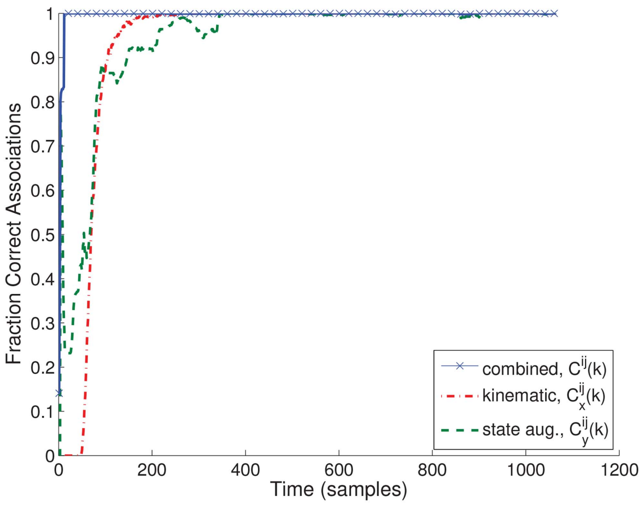


Fig. 19. Average association accuracy over 200 runs (scenario 2A), with window length $L_w = 200$.

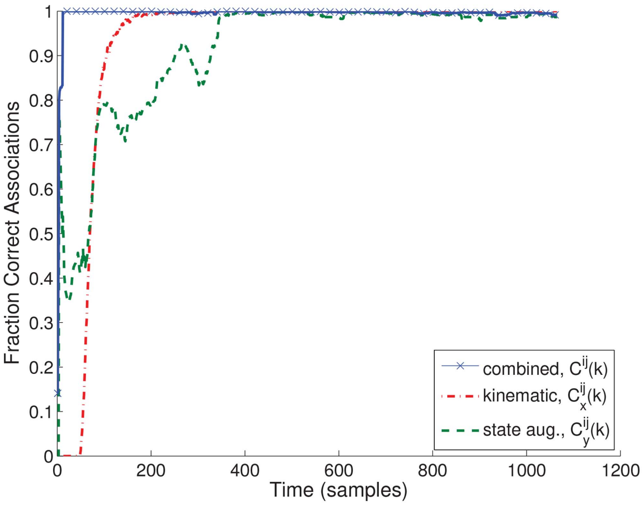


Fig. 20. Average association accuracy over 200 runs (scenario 2B), with window length $L_w = 200$.

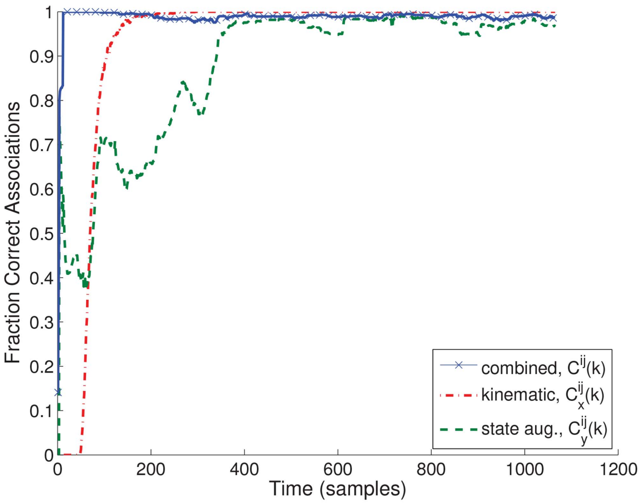


Fig. 21. Average association accuracy over 200 runs (scenario 2C), with window length $L_w = 200$.

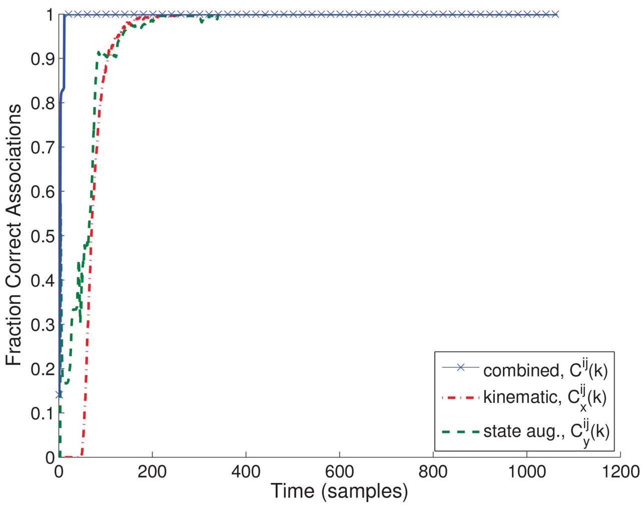


Fig. 22. Average association accuracy over 200 runs with unscented state augmentation costs (scenario 2A), window length $L_w = 200$.

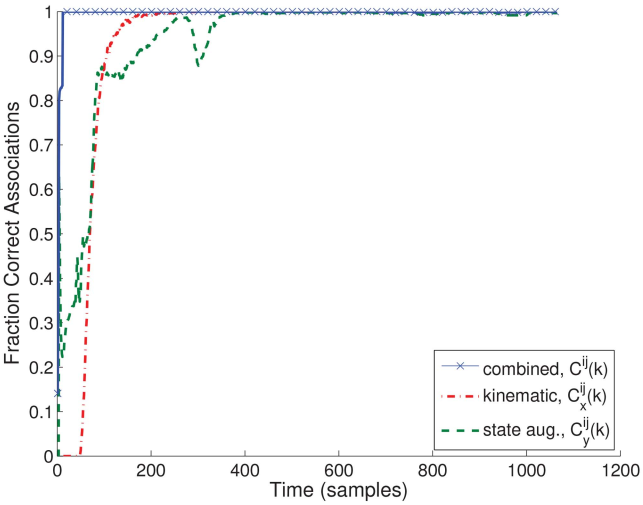


Fig. 23. Average association accuracy over 200 runs with unscented state augmentation costs (scenario 2B), window length $L_w = 200$.

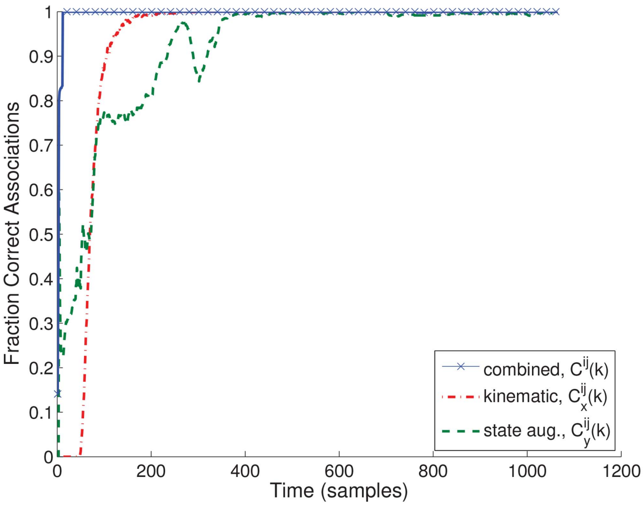


Fig. 24. Average association accuracy over 200 runs with unscented state augmentation costs (scenario 2C), window length $L_w = 200$.

TABLE III
Run Time Comparison of Kinematic Calculations vs. State Augmentation Calculations

	State Augmentation (s)	Kinematic (s)
Direct (1 pt)	2.781	6.003
Unscented (11 pts)	21.597	5.923

A comparison of the computer run time for the kinematic and state augmentation cost calculations for the scenario above is provided in Table III. The simulations were preformed using Matlab 2010b on an Intel Core2 Duo 2.66 GHz processor. The state augmentation cost calculations take longer in the case of the unscented transform since portions of the calculations must be performed at the 11 points used in the transform. The computational complexity of the state augmentation cost will, in general, depend largely on the nature of the state augmentation vector (size, nonlinearities, etc.).

Overall, the best T2TA accuracy is achieved by using the combined cost of association with the unscented transform modification of Section 3.3 to account for the uncertainty in the estimated state augmentation.

7. CONCLUSIONS

Performing T2TA between sensors is necessary to utilize any previously determined information about the tracks of targets from sensors which tracked earlier portions of the target trajectories. Earlier and more accurate association of tracks is desired, as there may be limited time to perform tracking once handover is performed. By utilizing an augmented state of the tracks at the first sensor, T2TA can be performed at the second sensor by combining both kinematic and state augmentation data. Additionally, the need for estimating the same state augmentation of each target directly at the second sensor is circumvented by matching the measured reflectivity of each target at the second sensor with the state augmentation estimated by the first sensor. The main contribution is this use of two nonlinearly related state augmentations and a method of accounting for their uncertainties. The combined cost explicitly allows for complete assignment with different numbers of tracks at each sensor. The combined cost was also found to provide highly accurate association of tracks earlier in a simulated tracking scenario, provided the estimate of the state augmentation is not too inaccurate. The use of very inaccurate state augmentation variables clouds the picture by flattening the likelihoods, deteriorating the association accuracy.

REFERENCES

[1] Y. Bar-Shalom, S. S. Blackman, and R. J. Fitzgerald
Dimensionless score function for multiple hypothesis tracking.
IEEE Transactions on Aerospace and Electronic Systems, **43**, 1 (Jan. 2007), 392–400.

[2] Y. Bar-Shalom and H. Chen
Multisensor track-to-track association for tracks with dependent errors.
Journal of Advances in Information Fusion, **1**, 1 (July 2006), 3–14.

[3] 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.

[4] Y. Bar-Shalom, P. K. Willett, and X. Tian
Tracking and Data Fusion.
YBS Publishing, 2011.

[5] R. R. Bate, D. D. Muellar, and J. E. White
Fundamentals of Astrodynamics.
New York: Dover Publications, Inc., 1971.

[6] S. Deb, K. Pattipati, and Y. Bar-Shalom
A multisensor-multitarget data association algorithm for heterogeneous sensors.
IEEE Transactions on Aerospace and Electronic Systems, **29**, 2 (Apr. 1997), 560–568.

[7] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom
A generalized S-D assignment algorithm for multisensor-multitarget state estimation.
IEEE Transactions on Aerospace and Electronic Systems, **33**, 2 (Apr. 1997), 523–538.

[8] J. Ferry
XMAP: Track-to-track association with metric, feature, and target-type data.
In *Proceedings of the 9th International Conference on Information Fusion*, July 2006, pp. 1–8.

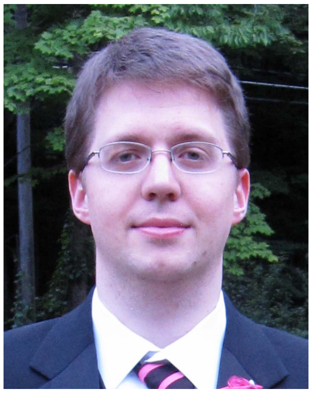
[9] S. J. Julier and J. K. Uhlmann
A new extension of the Kalman filter to nonlinear systems.
In *Proceedings of SPIE International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, Orlando, FL, 1997.

[10] X. R. Li and V. P. Jilkov
A survey of maneuvering target tracking—Part II: Ballistic target models.
In *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, San Diego, CA, Aug. 2001.

[11] M. Longbin, S. Xiaoquan, Z. Yiyu, S. Z. Kang, and Y. Bar-Shalom
Unbiased converted measurements for tracking.
IEEE Transactions on Aerospace and Electronic Systems, **34**, 3 (July 1998), 1023–1027.

[12] K. R. Pattipati, R. L. Popp, and T. Kirubarajan
Survey of assignment techniques for multitarget tracking.
In *Multitarget-Multisensor Tracking: Advances and Applications* vol. III, Y. Bar-Shalom and W. D. Blair, Eds., Dedham, MA: Artech House, 2000, ch. 2.

[13] R. L. Popp, K. R. Pattipati, and Y. Bar-Shalom
Dynamically adaptable m-best 2-D assignment algorithm and multilevel parallelization.
IEEE Transactions on Aerospace and Electronic Systems, **35**, 4 (Oct. 1999), 1145–1160.



Richard W. Osborne, III obtained his B.S. and M.S. degrees in electrical engineering from the University of Connecticut in 2004 and 2007, respectively.

He is a graduate student currently working on his Ph.D. in electrical engineering at the University of Connecticut, Storrs, CT. His academic interests include target tracking, data fusion, and other areas of estimation.

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, the 2008 IEEE D. J. Picard Medal for Radar Technologies and Applications, and the 2012 Connecticut Medal of Technology.

Peter Willett (F'03) received his B.A.Sc. (engineering science) from the University of Toronto in 1982, and his Ph.D. degree from Princeton University in 1986.

He has been a faculty member at the University of Connecticut since 1986, and since 1998 has been a professor. He has published 135 journal articles (13 more under review), 290 conference papers, and 9 book chapters. His primary areas of research have been statistical signal processing, detection, machine learning, data fusion and tracking. He has interests in and has published in the areas of change/abnormality detection, optical pattern recognition, communications and industrial/security condition monitoring.

He is editor-in-chief for *IEEE Transactions on Aerospace and Electronic Systems*, and until recently was associate editor for three active journals—*IEEE Transactions on Aerospace and Electronic Systems* (for Data Fusion and Target Tracking) and *IEEE Transactions on Systems, Man, and Cybernetics*, parts A and B. He is also associate editor for the IEEE AES Magazine, editor of the AES Magazine's periodic Tutorial issues, associate editor for ISIF's electronic *Journal of Advances in Information Fusion*, and is a member of the editorial board of IEEE's Signal Processing Magazine. He was a member of the IEEE AESS Board of Governors 2003–2009. He was general cochair (with Stefano Coraluppi) for the 2006 ISIF/IEEE Fusion Conference in Florence, Italy, Program Co-Chair (with Eugene Santos) for the 2003 IEEE Conference on Systems, Man & Cybernetics in Washington, D.C., and program cochair (with Pramod Varshney) for the 1999 Fusion Conference in Sunnyvale. He was coorganizer of the tracking subsession at the 1999 IEEE Aerospace Conference, and has been organizer of the Remote Sensing Track of that conference 2000–2003. Jointly with T. Kirubarajan he has coorganized the SPIE "System Diagnosis and Prognosis: Security and Condition Monitoring Issues" Conference in Orlando, 2001–2003. He has been a member of the IEEE Signal Processing Society's Sensor-Array & Multichannel (SAM) Technical Committee since 1997, and both serves on that TC's SAM Conferences' Program Committees and maintains the SAM website.

Posterior Cramér-Rao Bounds for Doppler Biased Distributed Tracking

XIUFENG SONG
PETER WILLETT
SHENGLI ZHOU

This paper investigates distributed tracking with *range-Doppler coupling*, where a range measurement of a target of interest is linearly biased by its range-rate (or Doppler). The coupling parameter λ can be zero, positive, or negative. The posterior Cramér-Rao bound (PCRB) is derived for distributed radar systems: multistatic and multiple-input multiple-output (MIMO) settings. In the multistatic case, a positive λ leads to the lowest PCRB, the same as is true for monostatic tracking. The paper also compares the tracking performance of multistatic and MIMO configurations, where the latter utilizes two waveforms with $\pm\lambda$ parameters, respectively. Regarding the power-unlimited case, a MIMO radar can always outperform a multistatic one from a tracking perspective. However, if the total power is limited, the situation is somewhat different: the transmitter co-located configuration is worse than a multistatic one, while in the widely-separated case the better choice depends on geometry.

Manuscript received August 31, 2011; revised October 27, 2011; released January 12, 2012.

Refereeing of this contribution was handled by Huimin Chen.

This work was supported by the U.S. Office of Naval Research under Grants N00014-07-10429 and N00014-09-10613.

Authors' address: Department of Electrical and Computer Engineering, University of Connecticut, 371 Fairfield Way Unit 2157, Storrs, CT 06269, E-mail: (xiufeng.song@gmail.com; willett@engr.uconn.edu; shengli@engr.uconn.edu).

1557-6418/12/$17.00 © 2012 JAIF

1. INTRODUCTION

A radar receiver extracts the range information of a moving target with a matched filter, of which the output is a slice of the waveform ambiguity function (AF) at Doppler shift f_d instead of zero in the absence of noise [6]. As a consequence, mismatch occurs and detection will degrade. To combat performance degradation of matched filtering against unknown Doppler shifts, Doppler tolerant (or insensitive) waveforms (DTWs), such as linear frequency modulation (LFM), P3 and P4 codes [6], appeared via introducing a slowly decaying *ridge* to their AFs as shown in Fig. 1. The range-Doppler ridge enables the matched filter to produce a slightly lower peak amplitude so as to avoid detection failure, but at the expense of introducing range bias [2], [6], [14], [15]. This phenomenon is termed *range-Doppler coupling*, and it is a compromise between range accuracy and Doppler robust detection.

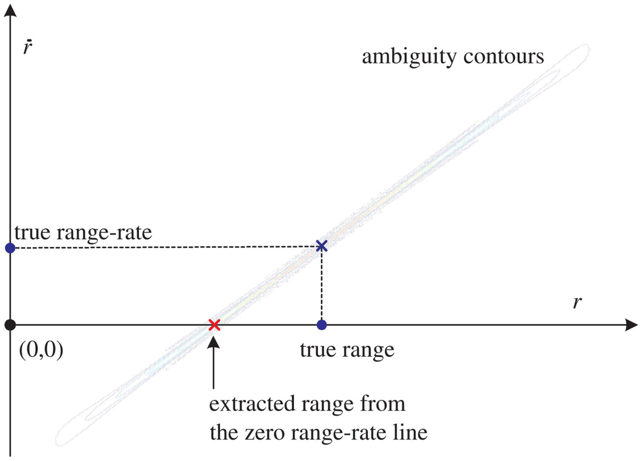


Fig. 1. An illustration of range extraction with a LFM. The target's true position is located at the center of the AF contours in (range, range-rate) plane. The matched filter extracts its range information from zero-range-rate axis; therefore, the measurement will be linearly *biased* by the target range-rate.

In some range-only tracking investigations, the observation is idealized to be true range plus a white Gaussian noise with known distribution. The characteristics of the waveform are neglected. As opposed to the ideal measurement model, the tracking problem is considered from a *system-level* in this paper. The radar system is assumed to utilize a DTW such as LFM to observe a target of interest; as a consequence, the extracted range is actually biased by range-rate (or Doppler). As [2] and [14], the range bias is assumed *linearly* related to the range-rate, with a constant range-Doppler coupling parameter.[1] The uniqueness of DTWs requires a proper mod-

[1] The range bias of a DTW depends on the shape of the decaying ridge of its AF. As for LFM, P3, and P4 waveforms, the ridge is like a tilted line through the zero-delay and zero-Doppler point in the two dimensional AF contour graph [6], and the linear bias modeling in general holds true [2] [14]. A numerical verification of its approximation accuracy can be found in [15].

ification in tracker design; direct application of classic approaches for such biased observations will to some degree degrade tracking performance.

One-dimensional tracking with Doppler biased range measurements has been investigated for a monostatic radar [2], [14]. In [2], Fitzgerald analyzed the measurement accuracy of DTWs, and showed that significant performance loss shall happen if a tracker mistook a biased range observation for an unbiased one. In [14], a steady state α-β filter is formulated for the DTWs, and the stationary estimate covariance is explored against variation of the tracking maneuvering index. In this paper, Doppler biased tracking is extended to distributed configurations including multistatic and multiple-input multiple-output (MIMO) systems. Do the same conclusions hold in the distributed case?

In [2] and [14], the target is assumed to have a constant range-rate; therefore, both the dynamic and measurement equations are linear. For this problem, the Kalman filter is the best, and the tracking potentials of different DTWs could be easily evaluated via a comparison of their stationary estimation results. Unfortunately, the distributed configuration is nonlinear and geometry dependent. No stationary solution is available. Instead of algorithm investigation, we are interested in the performance limitation of DTWs in target tracking, and a Cramér-Rao bound based study is adopted [12], [13]. In this paper, we make the following contributions:

- We model the range-Doppler coupling in two dimensions, and the geometric relationship between the time-varying range and range-rate is derived.
- We give the closed form Posterior Cramér-Rao bounds (PCRB) of Doppler biased tracking for a multistatic configuration.
- We analyze the *tracking* performance of different DTWs respectively with positive, negative, and zero coupling parameters. Numerical results show that the DTW with the positive coupling has lowest bound, while that with a negative one has the highest.
- We investigate the tracking performance of a two-transmitter MIMO radar, of which the waveforms are two DTWs respectively with positive and negative coupling parameters. If the system is power unlimited, a MIMO setup, either co-located or widely-separated, always outperforms the multistatic one. As for the power-limited scenario, the transmitter co-located setup is worse than the multistatic one utilizing DTW with positive coupling parameter, while whether a widely-separated MIMO is better than a multistatic case depends on the geometry.

The present paper collects the multistatic tracking conclusions from [10]; however, significant extensions including the MIMO setups are made. The rest of this paper is as follows: Section 2 introduces the range-Doppler coupling problem; Section 3 gives the tracking model; the PCRB is derived in Section 4; Section 6 analyzes the PCRB of a multistatic radar system, while Section 7 focuses on the performance of the PCRBs for different MIMO configurations; conclusions are drawn after that.

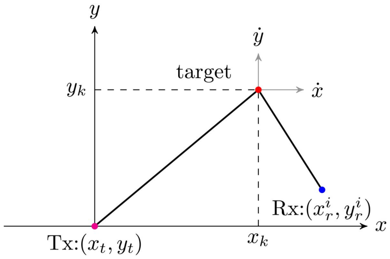


Fig. 2. The bistatic geometry: the coordinates of the transmitter Tx and the ith receiver Rx are respectively located at (x_t, y_t) and (x_r^i, y_r^i), while the target location, $(x_k + \dot{x}t, y_k + \dot{y}t)$, varies linearly with time t due to the constant velocity components projected on either axis.

2. PROBLEM STATEMENT

This paper studies the tracking performance of DTWs, of which the extracted range z of a target of interest is linearly biased by its range-rate $\dot{r}$ [2], [14], [15]

$$z = r + \lambda\dot{r} + n \tag{1}$$

where n denotes additive zero-mean white Gaussian noise, and λ stands for the waveform dependent *range-Doppler coupling parameter*. Generally, λ is known constant for a fixed waveform, and it can be zero, positive, or negative. For example, the coupling parameters of up- and down-sweep LFM waveforms are respectively expressed as $\lambda = f_c\tau/B$ and $\lambda = -f_c\tau/B$ [2], where f_c denotes the carrier frequency, τ is the pulse width, and B stands for the bandwidth.

Firstly, the bistatic range and range-rate are investigated for a given receiver i at the kth pulse repetition period (PRP). Let the coordinates of the transmitter, the ith receiver, and the target initial position at the kth PRP be (x_t, y_t), (x_r^i, y_r^i), and (x_k, y_k), respectively, as in Fig. 2. Then the corresponding bistatic time-varying range $r_k^i(t)$ is

$$\begin{aligned} r_k^i(t) &= \sqrt{(x_k + \dot{x}t - x_r^i)^2 + (y_k + \dot{y}t - y_r^i)^2} \\ &\quad + \sqrt{(x_k + \dot{x}t - x_t)^2 + (y_k + \dot{y}t - y_t)^2} \end{aligned} \tag{2}$$

where $\dot{x}$ and $\dot{y}$ respectively denotes the target velocity components along x and y axes, and $0 \leq t \leq \tau$. Suppose that $\dot{x}$ and $\dot{y}$ remain stationary within a single pulse width; the bistatic range-rate $\dot{r}_k^i(t)$ is written as

$$\begin{aligned} \dot{r}_k^i(t) &= \frac{(x_k + \dot{x}t - x_r^i)\dot{x} + (y_k + \dot{y}t - y_r^i)\dot{y}}{\sqrt{(x_k + \dot{x}t - x_r^i)^2 + (y_k + \dot{y}t - y_r^i)^2}} \\ &\quad + \frac{(x_k + \dot{x}t - x_t)\dot{x} + (y_k + \dot{y}t - y_t)\dot{y}}{\sqrt{(x_k + \dot{x}t - x_t)^2 + (y_k + \dot{y}t - y_t)^2}}. \end{aligned} \tag{3}$$

Obviously, both the bistatic range $r_k^i(t)$ and range-rate $\dot{r}_k^i(t)$ are time-varying. Assume that the target does not undergo significant spatial shift within a pulse width; thus $r_i(t)$ and $\dot{r}_i(t)$ can be approximated with the individual initial value:

$$\begin{aligned} r_k^i(t) &\approx r_k^i(0) \triangleq r_k^i = d_r^i(x_k,y_k) + d_t(x_k,y_k) \\ \dot{r}_k^i(t) &\approx \dot{r}_k^i(0) \triangleq \dot{r}_k^i = c_x^i(x_k,y_k)\dot{x} + c_y^i(x_k,y_k)\dot{y} \end{aligned} \tag{4}$$

where

$$\begin{aligned} d_r^i(x_k,y_k) &\triangleq \sqrt{(x_k - x_r^i)^2 + (y_k - y_r^i)^2} \\ d_t(x_k,y_k) &\triangleq \sqrt{(x_k - x_t)^2 + (y_k - y_t)^2} \end{aligned} \tag{5}$$

stand for the distances from the target's initial location to the ith receiver and the transmitter, respectively, and

$$\begin{aligned} c_x^i(x_k,y_k) &= \frac{x_k - x_r^i}{d_r^i(x_k,y_k)} + \frac{x_k - x_t}{d_t(x_k,y_k)} \\ c_y^i(x_k,y_k) &= \frac{y_k - y_r^i}{d_r^i(x_k,y_k)} + \frac{y_k - y_t}{d_t(x_k,y_k)}. \end{aligned} \tag{6}$$

Clearly, both r_k^i and $\dot{r}_k^i$ are geometry dependent.

3. TRACKING MODEL

In this section, multistatic tracking is investigated in two dimensions. The coordinates of the target of interest and its corresponding velocity components along the x-axis and y-axis are chosen to compose the state vector

$$\mathbf{s}_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]^T \tag{7}$$

where k indicates the pulse index. A *discrete white noise acceleration model* [1] is employed, of which the dynamic equation is linear

$$\mathbf{s}_{k+1} = \mathbf{F}\mathbf{s}_k + \boldsymbol{\Gamma}\mathbf{v}_k \tag{8}$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

is the time invariant system matrix, T_s denotes the PRP,

$$\boldsymbol{\Gamma} = \begin{bmatrix} T_s^2/2 & 0 \\ 0 & T_s^2/2 \\ T_s & 0 \\ 0 & T_s \end{bmatrix} \tag{10}$$

denotes the noise gain matrix, and

$$\mathbf{v}_k = [v_x, v_y]^T \tag{11}$$

is the additive process noise vector. In this paper, v_x and v_y are assumed independently and identically distributed (IID) zero-mean white Gaussian noise with variance σ_v^2; as a result, we have $\boldsymbol{\Gamma}\mathbf{v}_k \sim \mathcal{N}(\mathbf{0},\mathbf{Q})$, where

$$\mathbf{Q} = \sigma_v^2 \boldsymbol{\Gamma}\boldsymbol{\Gamma}^T = \sigma_v^2 \begin{bmatrix} T_s^4/4 & 0 & T_s^3/2 & 0 \\ 0 & T_s^4/4 & 0 & T_s^3/2 \\ T_s^3/2 & 0 & T_s^2 & 0 \\ 0 & T_s^3/2 & 0 & T_s^2 \end{bmatrix} \tag{12}$$

refers to the process noise (target maneuver), which under this model is singular.

Let the multistatic radar system be comprised of a single transmitter and N distributed receivers, all properly synchronized. Each receiver analyzes its individually collected echoes of a certain PRP to estimate the corresponding bistatic range. The range extracted by a matched filter is biased by the bistatic range-rate, and it is written as

$$z_k^i = h_i(\mathbf{s}_k) + w_i \tag{13}$$

for the ith receiver at the kth pulse, where

$$\begin{aligned} h_i(\mathbf{s}_k) = d_r^i(x_k,y_k) + d_t(x_k,y_k) + \lambda c_x^i(x_k,y_k)\dot{x}_k \\ + \lambda c_y^i(x_k,y_k)\dot{y}_k \end{aligned} \tag{14}$$

integrates the range-Doppler coupled bistatic geometry, and w_is stand for the additive IID white Gaussian noise with zero-mean and variance σ_r^2. Defining

$$\begin{aligned} \mathbf{z}_k &= [z_k^1, z_k^2, \ldots, z_k^N]^T \\ \mathbf{h}_k(\mathbf{s}_k) &= [h_1(\mathbf{s}_k), h_2(\mathbf{s}_k), \ldots, h_N(\mathbf{s}_k)]^T \\ \mathbf{w}_k &= [w_1, w_2, \ldots, w_N]^T \end{aligned} \tag{15}$$

the measurement equation is compactly expressed as

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{s}_k) + \mathbf{w}_k \tag{16}$$

where $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \sigma_r^2 \mathbf{I}_N)$, and $\mathbf{I}_N$ indicates the identity matrix with size $N \times N$.

Compared with an unbiased measurement equation as in [1], (16) contains two Doppler (range-rate) dependent items, $\lambda c_x^i(x_k,y_k)\dot{x}_k$s and $\lambda c_y^i(x_k,y_k)\dot{y}_k$s, to compensate the range bias. This slight modeling modification may significantly improve tracking performance with Doppler biased range measurements [2]. In addition, the measurement model (16) includes the unbiased situation as a special case when $\lambda = 0$.

4. POSTERIOR CRAMÉR-RAO BOUNDS

A. Background

Let $\hat{\boldsymbol{\theta}}$ be the estimate of a random vector $\boldsymbol{\theta}$ based on observation $\boldsymbol{\beta}$. Then the posterior (Bayesian) Cramér-Rao bound (PCRB) for the error covariance matrix satisfies [13]

$$\mathbf{C} \triangleq \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\beta}}\{(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T\} \succeq \mathbf{J}^{-1} \tag{17}$$

where $\mathbf{J}$ denotes the Bayesian information matrix (BIM), which is assumed to exist and be invertible, while the matrix inequality indicates that $(\mathbf{C} - \mathbf{J}^{-1})$ is positive semidefinite. Let

$$\nabla_{\mathbf{a}} = \left[\frac{\partial}{\partial a_1}, \frac{\partial}{\partial a_2}, \ldots, \frac{\partial}{\partial a_n}\right]^T \qquad \text{and} \qquad \Delta_{\mathbf{b}}^{\mathbf{a}} = \nabla_{\mathbf{b}} \nabla_{\mathbf{a}}^T \tag{18}$$

be operators of the first and second-order partial derivatives; hence, $\mathbf{J}$ is written as [13]

$$\mathbf{J} = \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\beta}}\{-\Delta_{\boldsymbol{\theta}}^{\boldsymbol{\theta}} \ln p(\boldsymbol{\theta},\boldsymbol{\beta})\} \tag{19}$$

where $p(\boldsymbol{\theta},\boldsymbol{\beta})$ denotes the joint probability density function (pdf) of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$.

For a tracking problem, the parameter vector and observation vector are respectively the collection of states and measurements

$$\boldsymbol{\theta}_k = [\mathbf{s}_1^T, \mathbf{s}_2^T, \ldots, \mathbf{s}_k^T]^T \qquad \text{and} \qquad \boldsymbol{\beta}_k = [\mathbf{z}_1^T, \mathbf{z}_2^T, \ldots, \mathbf{z}_k^T]^T \tag{20}$$

of which the vector sizes depend on the number of pulses. Based on (17), the estimation covariance for $\mathbf{s}_k$ is bounded by the right-lower block of $\mathbf{J}^{-1}$

$$\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\beta}}\{(\hat{\mathbf{s}}_k - \mathbf{s}_k)(\hat{\mathbf{s}}_k - \mathbf{s}_k)^T\} \succeq [\mathbf{J}^{-1}]_{(4k-3):4k,(4k-3):4k} \tag{21}$$

where $[\mathbf{J}^{-1}]_{(4k-3):4k,(4k-3):4k}$ denotes a submatrix of $\mathbf{J}^{-1}$ spanned with proper elements. Partition $\mathbf{J}$ into blocks

$$\mathbf{J} = \begin{bmatrix} [\mathbf{J}]_{1:4k-1,1:4k-4} & [\mathbf{J}]_{1:4k-4,(4k-3):4k} \\ [\mathbf{J}]_{(4k-3):4k,1:4k-4} & [\mathbf{J}]_{(4k-3):4k,(4k-3):4k} \end{bmatrix} \tag{22}$$

and then we obtain

$$[\mathbf{J}^{-1}]_{(4k-3):4k,(4k-3):4k} = \mathbf{J}_k^{-1}(\lambda) \tag{23}$$

where

$$\begin{aligned}\mathbf{J}_k(\lambda) = {} & [\mathbf{J}]_{(4k-3):4k,(4k-3):4k} - [\mathbf{J}]_{(4k-3):4k,1:4k-4} \\ & \cdot [\mathbf{J}]_{1:4k-1,1:4k-4}^{-1} [\mathbf{J}]_{1:4k-4,(4k-3):4k}\end{aligned} \tag{24}$$

is termed as the *Bayesian information submatrix* (BISM) for the state vector $\mathbf{s}_k$ [12], and λ emphasizes its waveform dependence. With the Markovian assumption

$$p(\boldsymbol{\theta}_k,\boldsymbol{\beta}_k) = p(\boldsymbol{\theta}_{k-1},\boldsymbol{\beta}_{k-1}) p(\mathbf{s}_k \mid \mathbf{s}_{k-1}) p(\mathbf{z}_k \mid \mathbf{s}_k) \tag{25}$$

$\mathbf{J}_k(\lambda)$ can be recursively calculated via the following lemma.

LEMMA 1 *The sequence* $\mathbf{J}_k$*s of BISMs for the estimate of state vectors* $\mathbf{s}_k$*s satisfy the recursion*

$$\mathbf{J}_{k+1}(\lambda) = \mathbf{D}_k^{22} - \mathbf{D}_k^{21}(\mathbf{J}_k(\lambda) + \mathbf{D}_k^{11})^{-1} \mathbf{D}_k^{12} \tag{26}$$

where

$$\begin{aligned}\mathbf{D}_k^{11} &= \mathbb{E}\{-\Delta_{\mathbf{s}_k}^{\mathbf{s}_k} \ln p(\mathbf{s}_{k+1} \mid \mathbf{s}_k)\} \\ \mathbf{D}_k^{12} &= \mathbb{E}\{-\Delta_{\mathbf{s}_k}^{\mathbf{s}_{k+1}} \ln p(\mathbf{s}_{k+1} \mid \mathbf{s}_k)\} = (\mathbf{D}_k^{21})^T \\ \mathbf{D}_k^{22} &= \mathbb{E}\{-\Delta_{\mathbf{s}_{k+1}}^{\mathbf{s}_{k+1}} \ln p(\mathbf{s}_{k+1} \mid \mathbf{s}_k)\} \\ & \quad + \mathbb{E}\{-\Delta_{\mathbf{s}_{k+1}}^{\mathbf{s}_{k+1}} \ln p(\mathbf{z}_{k+1} \mid \mathbf{s}_{k+1})\}.\end{aligned} \tag{27}$$

PROOF Proof can be found in [12].

B. PCRB Specification

Based on (8) and (16), $p(\mathbf{s}_{k+1} \mid \mathbf{s}_k)$ and $p(\mathbf{z}_k \mid \mathbf{s}_k)$ are both Gaussian and respectively with conditional pdfs

$$\begin{aligned}p(\mathbf{s}_{k+1} \mid \mathbf{s}_k) &= \frac{1}{(2\pi)^2 |\mathbf{Q}|^{1/2}} \\ & \quad \cdot \exp\left[-(1/2)(\mathbf{s}_{k+1} - \mathbf{F}\mathbf{s}_k)^T \mathbf{Q}^{-1} (\mathbf{s}_{k+1} - \mathbf{F}\mathbf{s}_k)\right] \\ p(\mathbf{z}_k \mid \mathbf{s}_k) &= \frac{1}{\left(\sqrt{2\pi}\sigma_r\right)^N} \exp\left[-(1/2\sigma_r^2)\|\mathbf{z}_k - \mathbf{h}_k(\mathbf{s}_k)\|^2\right]\end{aligned} \tag{28}$$

where $|\mathbf{Q}|$ denotes the determinant of $\mathbf{Q}$ and $\|\cdot\|$ denotes the l_2 norm. It is easy to verify that

$$\begin{aligned}\mathbf{D}_k^{11} &= \mathbf{F}^T \mathbf{Q}^{-1} \mathbf{F} \\ \mathbf{D}_k^{12} &= -\mathbf{F}^T \mathbf{Q}^{-1} \\ \mathbb{E}\{-\Delta_{\mathbf{s}_{k+1}}^{\mathbf{s}_{k+1}} \ln p(\mathbf{s}_{k+1} \mid \mathbf{s}_k)\} &= \mathbf{Q}^{-1}.\end{aligned} \tag{29}$$

Due to the nonlinearity of $\mathbf{h}_k(\mathbf{s}_k)$, the second term of $\mathbf{D}_k^{22}$ is not straightforward. Fortunately, with the fact that $p(\mathbf{s}_k, \mathbf{z}_k) = p(\mathbf{s}_k) p(\mathbf{z}_k \mid \mathbf{s}_k)$, we have

$$\mathbb{E}\{-\Delta_{\mathbf{s}_k}^{\mathbf{s}_k} \ln p(\mathbf{z}_k \mid \mathbf{s}_k)\} = \mathbb{E}_{\mathbf{s}_k}\{\mathbf{P}_k(\lambda)\} \tag{30}$$

where

$$\mathbf{P}_k(\lambda) \triangleq \mathbb{E}_{\mathbf{z}_k|\mathbf{s}_k}\{-\Delta_{\mathbf{s}_k}^{\mathbf{s}_k} \ln p(\mathbf{z}_k \mid \mathbf{s}_k)\} \tag{31}$$

is a standard Fisher information matrix (FIM) for $\mathbf{z}_k$ [1], [13]. Substituting $p(\mathbf{z}_k \mid \mathbf{s}_k)$ into (31), we have

$$\begin{aligned}\mathbf{P}_k(\lambda) &= \frac{1}{2\sigma_r^2} \sum_{i=1}^N \mathbb{E}_{\mathbf{z}_k|\mathbf{s}_k}\{\Delta_{\mathbf{s}_k}^{\mathbf{s}_k}(z_k^i - h_i(\mathbf{s}_k))^2\} \\ &= \frac{1}{\sigma_r^2} \sum_{i=1}^N \mathbb{E}_{\mathbf{z}_k|\mathbf{s}_k}\{\nabla_{\mathbf{s}_k}[(h_i(\mathbf{s}_k) - z_k^i)\nabla_{\mathbf{s}_k}^T h_i(\mathbf{s}_k)]\} \\ &= \frac{1}{\sigma_r^2} \sum_{i=1}^N \mathbb{E}_{\mathbf{z}_k|\mathbf{s}_k}\{\nabla_{\mathbf{s}_k} h_i(\mathbf{s}_k) \nabla_{\mathbf{s}_k}^T h_i(\mathbf{s}_k)\} \\ & \quad + \frac{1}{\sigma_r^2} \sum_{i=1}^N \Delta_{\mathbf{s}_k}^{\mathbf{s}_k} h_i(\mathbf{s}_k) \mathbb{E}_{\mathbf{z}_k|\mathbf{s}_k}\{h_i(\mathbf{s}_k) - z_k^i\}.\end{aligned} \tag{32}$$

Since $\mathbb{E}_{\mathbf{z}_k|\mathbf{s}_k}\{h_i(\mathbf{s}_k) - z_k^i\} = 0$, $\mathbf{P}_k(\lambda)$ can be simplified as

$$\mathbf{P}_k(\lambda) = \frac{1}{\sigma_r^2} \sum_{i=1}^N \nabla_{\mathbf{s}_k} h_i(\mathbf{s}_k) \nabla_{\mathbf{s}_k}^T h_i(\mathbf{s}_k) \tag{33}$$

where $\nabla_{\mathbf{s}_k} h_i(\mathbf{s}_k)$ is written as

$$\nabla_{\mathbf{s}_k} h_i(\mathbf{s}_k) = \left[\frac{\partial h_i(\mathbf{s}_k)}{\partial x_k}, \frac{\partial h_i(\mathbf{s}_k)}{\partial y_k}, \frac{\partial h_i(\mathbf{s}_k)}{\partial \dot{x}_k}, \frac{\partial h_i(\mathbf{s}_k)}{\partial \dot{y}_k}\right]^T. \tag{34}$$

In the following, the elements of $\nabla_{\mathbf{s}_k} h_i(\mathbf{s}_k)$ will be specified based on (14). Since

$$\begin{aligned}\frac{\partial}{\partial x_k}\{d_r^i(x_k,y_k)+d_t(x_k,y_k)\} &= c_x^i(x_k,y_k)\\ \frac{\partial}{\partial y_k}\{d_r^i(x_k,y_k)+d_t(x_k,y_k)\} &= c_y^i(x_k,y_k)\end{aligned} \tag{35}$$

we have

$$\begin{aligned}\frac{\partial h_i(\mathbf{s}_k)}{\partial \dot{x}_k} &= \lambda c_x^i(x_k,y_k)\\ \frac{\partial h_i(\mathbf{s}_k)}{\partial \dot{y}_k} &= \lambda c_y^i(x_k,y_k)\\ \frac{\partial h_i(\mathbf{s}_k)}{\partial x_k} &= c_x^i(x_k,y_k)+\lambda\dot{x}_k\frac{\partial c_x^i(x_k,y_k)}{\partial x_k}+\lambda\dot{y}_k\frac{\partial c_y^i(x_k,y_k)}{\partial x_k}\\ \frac{\partial h_i(\mathbf{s}_k)}{\partial y_k} &= c_y^i(x_k,y_k)+\lambda\dot{x}_k\frac{\partial c_x^i(x_k,y_k)}{\partial y_k}+\lambda\dot{y}_k\frac{\partial c_y^i(x_k,y_k)}{\partial y_k}\end{aligned} \tag{36}$$

where

$$\frac{\partial c_x^i(x_k,y_k)}{\partial x_k} = \frac{(y_k-y_r^i)^2}{(d_r^i(x_k,y_k))^3}+\frac{(y_k-y_t)^2}{d_t^3(x_k,y_k)} \tag{37}$$

$$\frac{\partial c_y^i(x_k,y_k)}{\partial y_k} = \frac{(x_k-x_r^i)^2}{(d_r^i(x_k,y_k))^3}+\frac{(x_k-x_t)^2}{d_t^3(x_k,y_k)} \tag{38}$$

$$\begin{aligned}\frac{\partial c_x^i(x_k,y_k)}{\partial y_k} &= \frac{\partial c_y^i(x_k,y_k)}{\partial x_k}\\ &= -\frac{(x_k-x_r^i)(y_k-y_r^i)}{(d_r^i(x_k,y_k))^3}-\frac{(x_k-x_t)(y_k-y_t)}{d_t^3(x_k,y_k)}.\end{aligned} \tag{39}$$

Substituting (36) into (33), $\mathbf{P}_k(\lambda)$ can be obtained. Finally, substituting (36) and (29) into (26), the recursion in *Lemma 1* is rewritten as

$$\mathbf{J}_{k+1}(\lambda) = [\mathbf{Q}+\mathbf{F}\mathbf{J}_k^{-1}(\lambda)\mathbf{F}^T]^{-1}+\mathbb{E}_{\mathbf{s}_{k+1}}\{\mathbf{P}_{k+1}(\lambda)\}. \tag{40}$$

The expectation towards $\mathbf{P}_{k+1}(\lambda)$ involves a complicated 4-fold integration. A closed form expression is elusive, hence Monte Carlo methods [3] are usually used.

When $\lambda = 0$, (40) degenerates to the case without range-Doppler coupling. Another special case is the noiseless kinematic model [1], where the state covariance is assumed to be zero, say $\mathbf{Q} = \mathbf{0}$. Hence, the expectation in (40) disappears, and the recursion is simplified to

$$\mathbf{J}_{k+1}(\lambda) = [\mathbf{F}\mathbf{J}_k^{-1}(\lambda)\mathbf{F}^T]^{-1}+\mathbf{P}_{k+1}(\lambda). \tag{41}$$

An implicit assumption behind (40) is that the radar system has perfect detection: the probability of detection is one, while that of false alarm is zero. Such an assumption obviously simplifies PCRB analysis; however, it may not hold true in some practical applications, particularly for those with low signal-to-noise ratio (SNR). An interesting discussion on PCRB with nonideal detection can be found in [8] and [16]; this paper contains no treatment of measurement origin uncertainty.

5. PCRB WITH CONSTANT ACCELERATION

Suppose that the target undergoes a constant acceleration. If its velocity significantly changes within a pulse duration τ, the linear bias model (1) would no longer hold true as the AF will be distorted [5]. In the following, $\ddot{x}\tau$ and $\ddot{y}\tau$ are assumed to be very small (actually, negligible), where $\ddot{x}$ and $\ddot{y}$ respectively denote the acceleration components along x and y axes. This is a fair assumption: for a practical radar pulse duration $\tau = 30$ μs, the velocity change is only 3×10^{-3} m/s even though the acceleration is as high as 100 m/s^2. The time-varying bistatic range for receiver i at the kth pulse is written as

$$\begin{aligned}\bar{r}_k^i(t) = {}&\sqrt{(x_k+\dot{x}t+0.5\ddot{x}t^2-x_r^i)^2+(y_k+\dot{y}t+0.5\ddot{y}t^2-y_r^i)^2}\\ &+\sqrt{(x_k+\dot{x}t+0.5\ddot{x}t^2-x_t)^2+(y_k+\dot{y}t+0.5\ddot{y}t^2-y_t)^2}\end{aligned} \tag{42}$$

where $0\le t\le\tau$. Since $\ddot{x}t^2 \ll \dot{x}t$ and $\ddot{y}t^2 \ll \dot{y}t$, we have

$$\begin{aligned}\bar{r}_k^i(t) \simeq {}&\sqrt{(x_k+\dot{x}t-x_r^i)^2+(y_k+\dot{y}t-y_r^i)^2}\\ &+\sqrt{(x_k+\dot{x}t-x_t)^2+(y_k+\dot{y}t-y_t)^2} = r_k^i(t).\end{aligned} \tag{43}$$

Therefore, the geometry results in Section 2 are still applicable for PCRB with a low acceleration.

The PRP is much larger than pulse width. The acceleration effect between pulses cannot be ignored in target tracking. Mathematically, the state vector should be modified as

$$\bar{\mathbf{s}}_k = [x_k,y_k,\dot{x}_k,\dot{y}_k,\ddot{x}_k,\ddot{y}_k]^T. \tag{44}$$

The dynamic equation falls into the *discrete Wiener process acceleration model* [1]

$$\bar{\mathbf{s}}_{k+1} = \bar{\mathbf{F}}\bar{\mathbf{s}}_k+\bar{\boldsymbol{\Gamma}}\mathbf{v}_k \tag{45}$$

where

$$\bar{\mathbf{F}} = \begin{bmatrix}1 & 0 & T_s & 0 & T_s^2/2 & 0\\ 0 & 1 & 0 & T_s & 0 & T_s^2/2\\ 0 & 0 & 1 & 0 & T_s & 0\\ 0 & 0 & 0 & 1 & 0 & T_s\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix} \tag{46}$$

and

$$\bar{\boldsymbol{\Gamma}} = \begin{bmatrix}T_s^2/2 & 0 & T_s & 0 & 1 & 0\\ 0 & T_s^2/2 & 0 & T_s & 0 & 1\end{bmatrix}^T. \tag{47}$$

Based on (43), the contribution of $\ddot{x}$ and $\ddot{y}$ to range is negligible; the measurement equation (14) remains unchanged. Using $\bar{\mathbf{F}}$, $\bar{\mathbf{G}}$, and $\bar{\mathbf{Q}} = \sigma_v^2\bar{\boldsymbol{\Gamma}}\bar{\boldsymbol{\Gamma}}^T$ to properly modify the corresponding parts in Section 4, the PCRB for constant acceleration will be obtained

$$\bar{\mathbf{J}}_{k+1}(\lambda) = [\bar{\mathbf{Q}} + \bar{\mathbf{F}}\bar{\mathbf{J}}_k^{-1}(\lambda)\bar{\mathbf{F}}^T]^{-1} + \mathbb{E}_{\bar{\mathbf{s}}_{k+1}}\{\bar{\mathbf{P}}_{k+1}(\lambda)\} \tag{48}$$

where

$$\begin{aligned}\bar{\mathbf{P}}_k(\lambda) &= \frac{1}{\sigma_r^2}\sum_{i=1}^{N} \nabla_{\bar{\mathbf{s}}_k} h_i(\bar{\mathbf{s}}_k)\nabla^T_{\bar{\mathbf{s}}_k} h_i(\bar{\mathbf{s}}_k)\\ &= \frac{1}{\sigma_r^2}\sum_{i=1}^{N} \nabla_{\bar{\mathbf{s}}_k} h_i(\mathbf{s}_k)\nabla^T_{\bar{\mathbf{s}}_k} h_i(\mathbf{s}_k)\end{aligned} \tag{49}$$

and $\nabla_{\bar{\mathbf{s}}_k} h_i(\mathbf{s}_k)$ is specified as

$$\begin{aligned}\nabla_{\bar{\mathbf{s}}_k} h_i(\mathbf{s}_k) &= \left[\frac{\partial h_i(\mathbf{s}_k)}{\partial x_k}, \frac{\partial h_i(\mathbf{s}_k)}{\partial y_k}, \frac{\partial h_i(\mathbf{s}_k)}{\partial \dot{x}_k}, \frac{\partial h_i(\mathbf{s}_k)}{\partial \dot{y}_k}, 0, 0\right]^T\\ &= [(\nabla_{\mathbf{s}_k} h_i(\mathbf{s}_k))^T, 0, 0]^T.\end{aligned} \tag{50}$$

6. λ: POSITIVE, NEGATIVE OR ZERO?

Were this treated in one dimension (range) the optimal tracker would be Kalman, and the stationary estimation covariance (from the Riccati equation) could reflect the performance of different waveforms. Here, however—in a distributed configuration and working in two Cartesian dimensions—the measurement equations are geometry dependent; no stationary solution exists, hence the performances for different λs are compared with the help of their PCRBs. Let the multistatic radar system be composed by one transmitter and four receivers, of which the coordinates are shown in Fig. 3. The PRP, receiver noise, and maneuvering index [1] are all fixed with values $T_s = 2$ s, $\sigma_r = 5$ m/s and $\kappa = \sigma_v T_s^2/\sigma_r = 1$. The PCRBs of position tracking errors

$$\sqrt{[\mathbf{J}_k^{-1}(\lambda)]_{1,1} + [\mathbf{J}_k^{-1}(\lambda)]_{2,2}} \tag{51}$$

for different λs. Firstly, we compare their tracking performance for a constant velocity target, of which the speed is 400 m/s. The target has four trajectories as shown in Fig. 3, and the calculated results for them are shown in Fig. 4. From those figures, we see that:

- The PCRBs are geometry dependent. The curves with different trajectories have quite dissimilar shapes. In addition, the curves with different λs but the same trajectory have similar shapes, but (slightly) different values.
- The bounds for positive λ are lower than those for a zero λ, while a negative λ introduces a higher bound. However, those of $|\lambda|$ and $-|\lambda|$ are not necessarily symmetric with respect to that of $\lambda = 0$.
- With the increase of λ, the bound becomes lower for a given trajectory. Note that λ is waveform dependent, and $|\lambda|$ cannot be arbitrarily large.

Secondly, the constant acceleration is considered. The target starts from stationary with an acceleration rate 15 m/s^2. Its four trajectories share the same initial points and directions as those in the previous example. Their PCRBs are illustrated in Fig. 5. Obviously, similar observations can be found as constant velocity case. Due to their similarity, we will not investigate the acceleration model in the MIMO tracking.

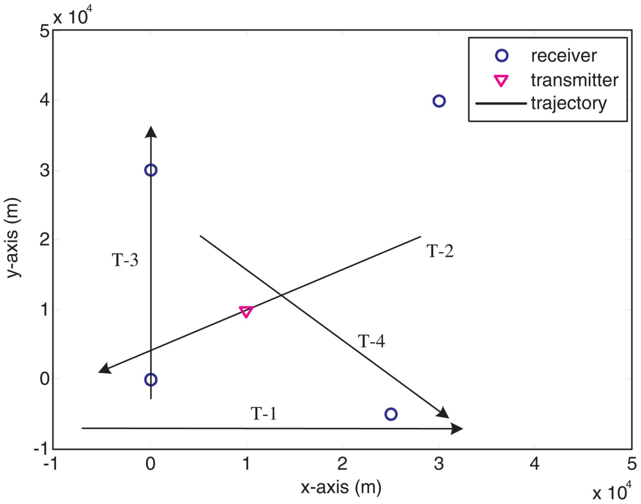


Fig. 3. An illustration of a multistatic constellation with a single transmitter and 4 receivers. Four typical target trajectories are used in the simulation: T-1 stands for a trajectory outside of the constellation, T-2 is the situation crossing the transmitter, T-3 stands for the case crossing the boundary receivers, while T-4 is a general one within the constellation.

To reemphasize that a positive λ leads to better tracking performance than a negative one is consistent with the theoretical analysis for the monostatic tracking [14]. An intuitive explanation is in Fig. 6. Since the range and range-rate are positively correlated in the state equation, their prior uncertainty can be considered as an ellipse centered at the range and range-rate of truth (the black dot), where the major axis of the ellipse has a positive slope. In the absence of noise, the two measurement extraction lines for waveforms, respectively with slope $\pm\lambda$, traverse the black dot; however, the contamination could slightly shifts them away from the noiseless situations in either direction (dashed lines). The possible shifts engender two measurement uncertainty bands (area between the two parallel dashed lines) as shown in Fig. 6. The uncertainty band for the positive λ and prior uncertainty ellipse share less overlap than that of the negative one. Consequently, a positive λ induces greater error cancelation in updating maintenance and result in better tracking performance.

Interestingly, waveforms with $\pm\lambda$ share the same measurement mean square errors, but with different PCRBs for a distributed configuration. This again verifies that *observations with the same measurement quality do not necessarily result in the same tracking performance* [9]. Moreover, a waveform with positive λ induces a lower bound than that with a zero λ, so the biased measurements are not always bad from the tracking viewpoint if the bias can be properly modeled.

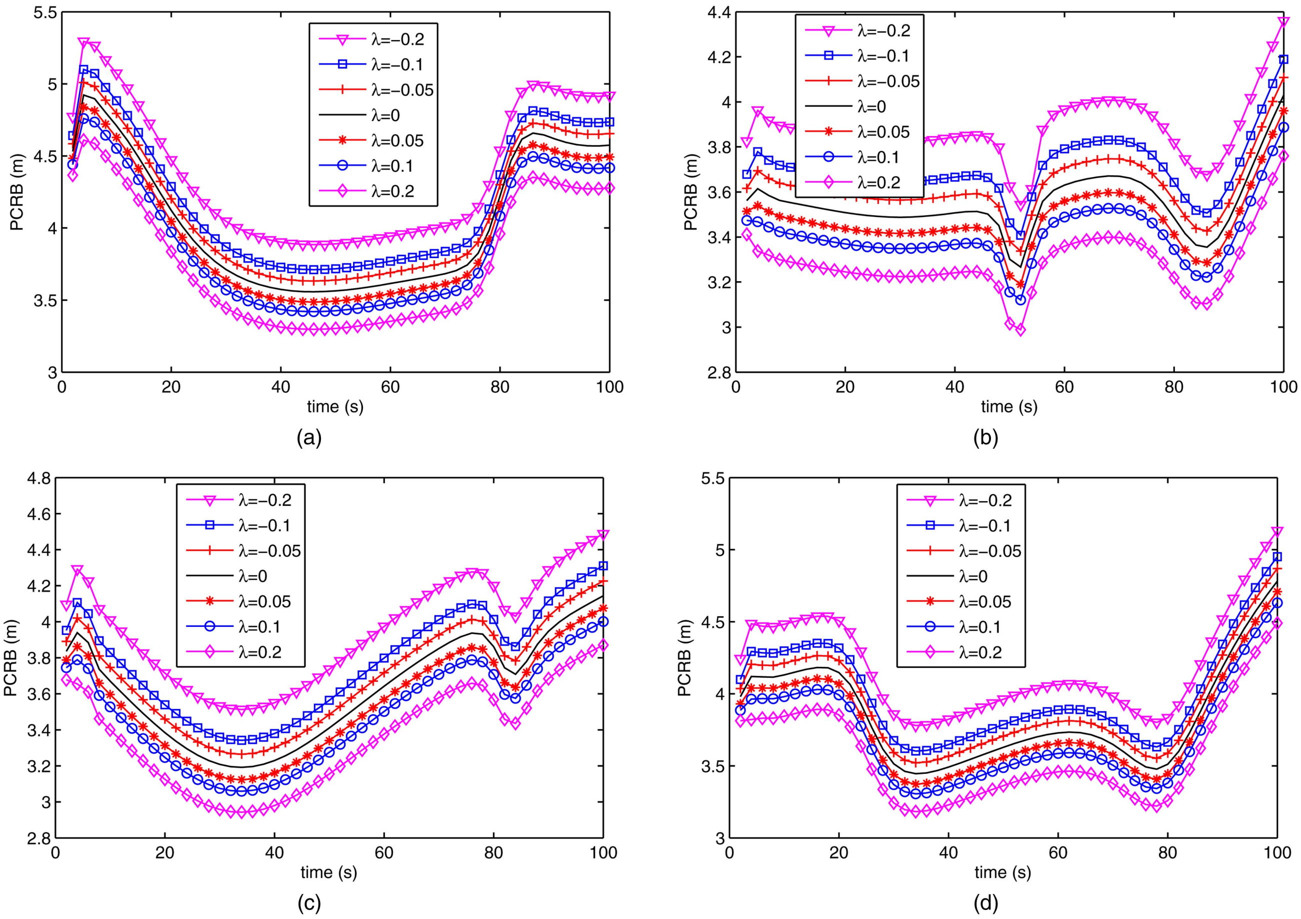


Fig. 4. PCRBs of a multistatic radar with different waveform parameters λs and target trajectories for constant velocity. The multistatic constellation and target trajectories are shown in Fig. 3. The prior distribution is $P_0 = \mathrm{diag}([20,20,5,5])$.

7. MIMO RADAR WAVEFORM COOPERATION

MIMO radar is an emerging concept employing waveform and spatial diversities to improve system performance [4], [7], [11]. Roughly, it can be divided into two categories based on the transmitter configuration: co-located [7] and widely-separated [4]. The multiple transmission trait requires the waveforms to be (nearly) orthogonal. Apparently the up- and down-sweep LFMs have very low cross-correlation even with moderate Doppler shift; therefore, they can be used in a two-transmitter MIMO system [17].

A. Power Unlimited System

Full power transmission is assumed to be performed at each antenna for this scenario. If the transmitters are homogeneous, the total radiated energy doubles as the number of transmitters N_t increases from one to two. Let the waveform energy keep constant; the range variance σ_r^2 remains the same for both multistatic and MIMO radar systems. Suppose the multistatic radar is the special case of the MIMO one via shutting down one transmitter—that is single-input multiple-output (SIMO)—and then we have the following results.

Lemma 2 *Let* $\mathbf{J}_k^{\mathrm{MU}}(\lambda,\sigma_r^2)$ *denote the BISM of a power unlimited MIMO radar with two waveforms parameterized with* $\pm\lambda$ *and noise variance* σ_r^2*, and let* $\mathbf{J}_k^{\mathrm{SU}}(\lambda,\sigma_r^2)$ *denote that of a multistatic one with proper parameters. Suppose* $\mathbf{J}_k^{\mathrm{MU}}(\lambda,\sigma_r^2) = \mathbf{J}_k^{\mathrm{SU}}(\lambda,\sigma_r^2)$*, and then we have* $\mathbf{J}_{k+1}^{\mathrm{MU}}(\lambda,\sigma_r^2) \succeq \mathbf{J}_{k+1}^{\mathrm{SU}}(\lambda,\sigma_r^2)$.

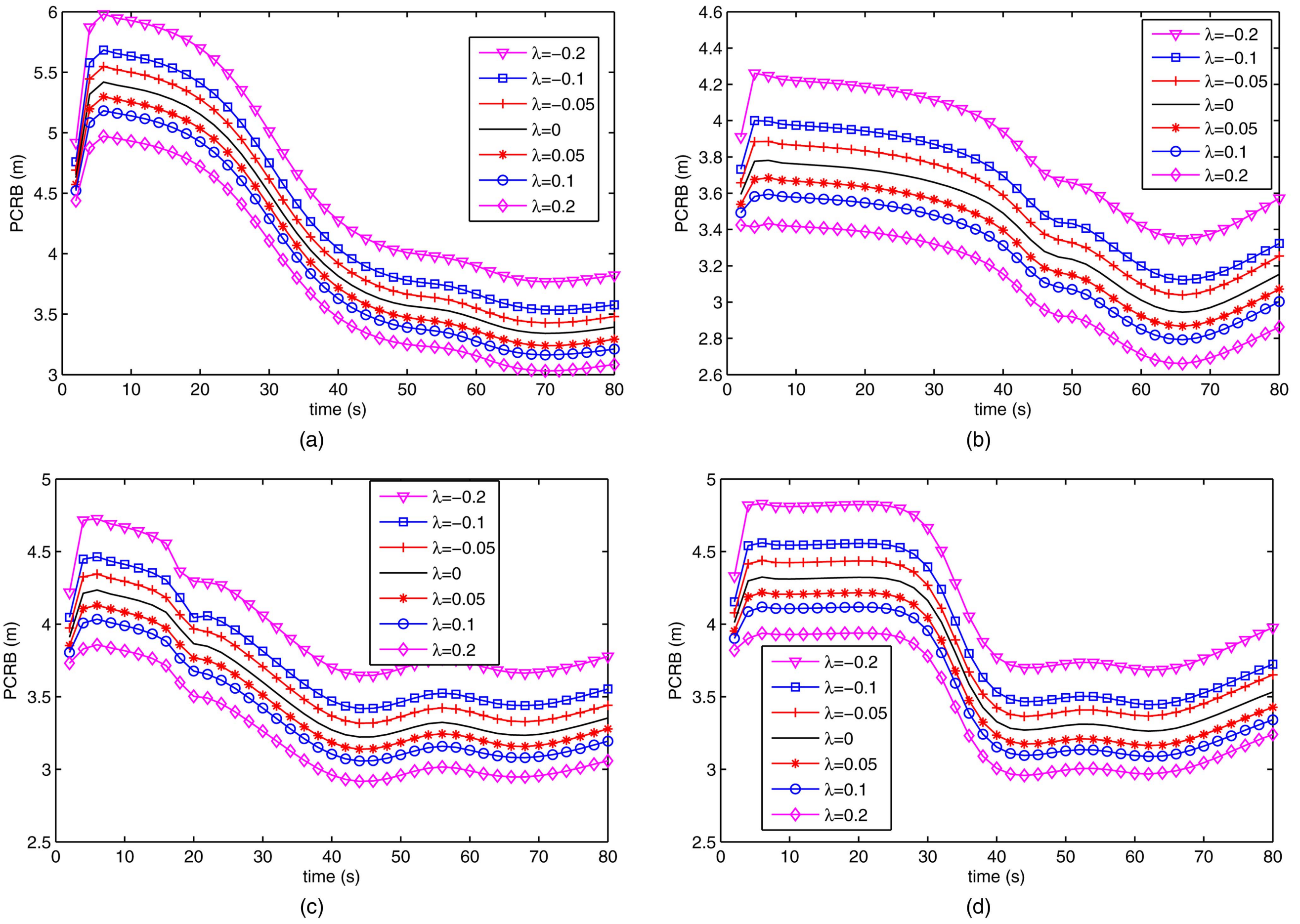
Proof Proof is straightforward.

In the power unlimited scenario, the MIMO radar is essentially composed of two multistatic ones. Apparently, it can obtain more information about the target, no matter whether the transmitters are co-located or widely-separated.

B. Power Limited System

The total energy is fixed for a power limited system; for simplicity, it is shared uniformly among transmitters. Based on (4) of [14], the measurement variance σ_r^2 is inversely proportional to the waveform power. So it is reasonable to assume that the range variance of the multistatic radar is half of that of a two-transmitter MIMO radar.

1) *Transmitter Co-located Case*: The co-located transmitter configuration does not enhance the spatial diversity. Based on the analysis in the previous section,



(a)

(b)

(c)

(d)

Fig. 5. PCRBs of a multistatic radar with different waveform parameters λs and target trajectories for constant acceleration. The prior distribution is $P_0 = \mathrm{diag}([20, 20, 5, 5, 1, 1])$.

MIMO is better than the multistatic case with a negative λ, but worse than that with a positive λ. Now, we are still interested in another problem: whether a MIMO radar is better than a multistatic radar with a zero λ. A thorough analysis would be complex; to simplify it, we adopt an assumption that the target is in the far-field of the receivers and transmitters; mathematically, $d_r^i(x_k, y_k)$ and $d_t^i(x_k, y_k)$ are both very large. Since the denominators of each item of $\partial c_x^i / \partial x_k$, $\partial c_y^i / \partial x_y$, and $\partial c_x^i / \partial x_y$ have higher orders than the numerators, we have

$$\frac{\partial c_x^i}{\partial x_k} \approx \frac{\partial c_y^i}{\partial y_k} \approx \frac{\partial c_x^i}{\partial y_k} \approx \frac{\partial c_y^i}{\partial x_k} \approx 0. \tag{52}$$

Moreover, the target speed is assumed low, and $|\lambda|$ is in general small; as a consequence, $\nabla_{\mathbf{s}_k} h_i(k)$ can be approximated as

$$\mathbf{g}_i(\lambda) \overset{\Delta}{=} \nabla_{\mathbf{s}_k} h_i(k) \approx [c_x^i, c_y^i, \lambda c_x^i, \lambda c_y^i]^T. \tag{53}$$

Substitute (53) into (33), the FIM is rewritten as

$$\bar{\mathbf{P}}_k(\lambda) = \frac{1}{\sigma_r^2} \sum_{i=1}^{N} \mathbf{g}_i(\lambda) \mathbf{g}_i^T(\lambda) = \frac{1}{\sigma_r^2} \begin{bmatrix} 1 & \lambda \\ \lambda & \lambda^2 \end{bmatrix} \otimes \mathbf{G} \tag{54}$$

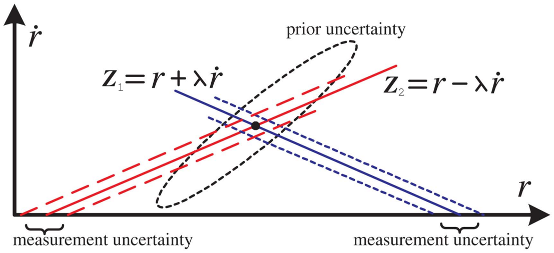
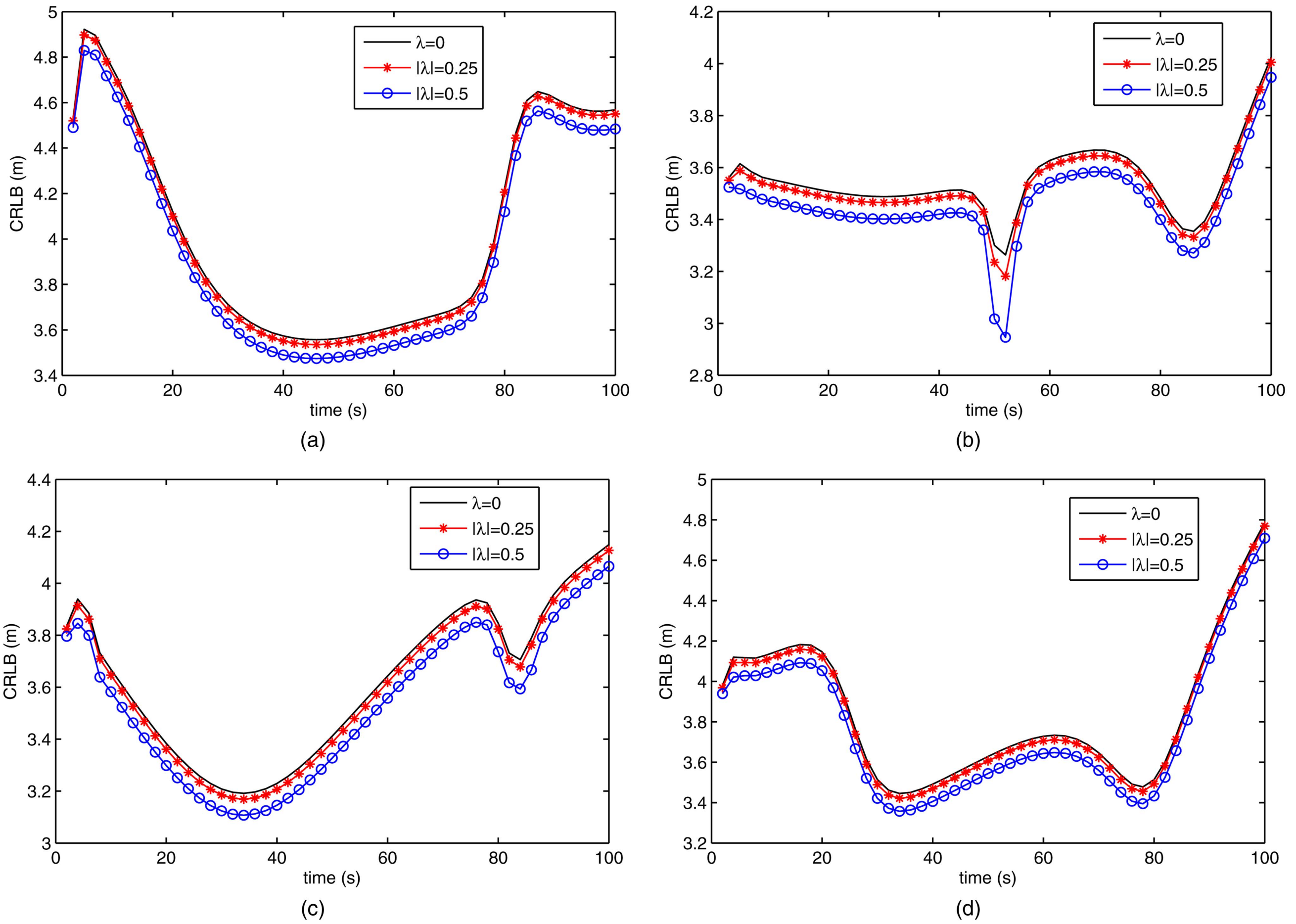


Fig. 6. An intuitive explanation of why a DTW with a positive coupling parameter outperforms that with a negative one in target tracking, where $\lambda > 0$ in this figure. Since the range r and range-rate $\dot{r}$ are positively correlated in the state equation, the prior uncertainty could be considered as an ellipse with positive major axis. If the coupling parameter is positive, the measurement and prior uncertainties will have less overlapping area. Therefore, it has better error cancelation capacity.

where

$$\mathbf{G} = \begin{bmatrix} \sum_{i=1}^{N} (c_x^i)^2 & \sum_{i=1}^{N} c_x^i c_y^i \\ \sum_{i=1}^{N} c_x^i c_y^i & \sum_{i=1}^{N} (c_y^i)^2 \end{bmatrix} \tag{55}$$

is a positive semidefinite matrix.



(a)

(b)

(c)

(d)

Fig. 7. PCRBs of a transmitter co-located MIMO radar with different waveform parameters λs and target trajectories. The prior distribution is $P_0 = \mathrm{diag}([20,20,5,5])$.

Lemma 3 *Let* $\mathbf{J}_k^{\mathrm{ML}}(\lambda,2\sigma_r^2)$ *and* $\mathbf{J}_k^{\mathrm{SU}}(\lambda,\sigma_r^2)$ *respectively denote the BISMs of a power limited co-located MIMO and multistatic radars with proper parameters. Suppose* $\mathbf{J}_k^{\mathrm{ML}}(\lambda,2\sigma_r^2) = \mathbf{J}_k^{\mathrm{SL}}(0,\sigma_r^2)$ *and (52) holds true, and then we have* $\mathbf{J}_{k+1}^{\mathrm{ML}}(\lambda,2\sigma_r^2) \succeq \mathbf{J}_{k+1}^{\mathrm{SL}}(0,\sigma_r^2)$.

Proof If (52) holds true, we have

$$\bar{\mathbf{P}}_{k+1}^{\mathrm{ML}}(\lambda) - \bar{\mathbf{P}}_{k+1}^{\mathrm{SL}}(0) = \frac{2}{\sigma_r^2}\mathrm{Diag}([0,\lambda^2]) \otimes \mathbf{G} \qquad (56)$$

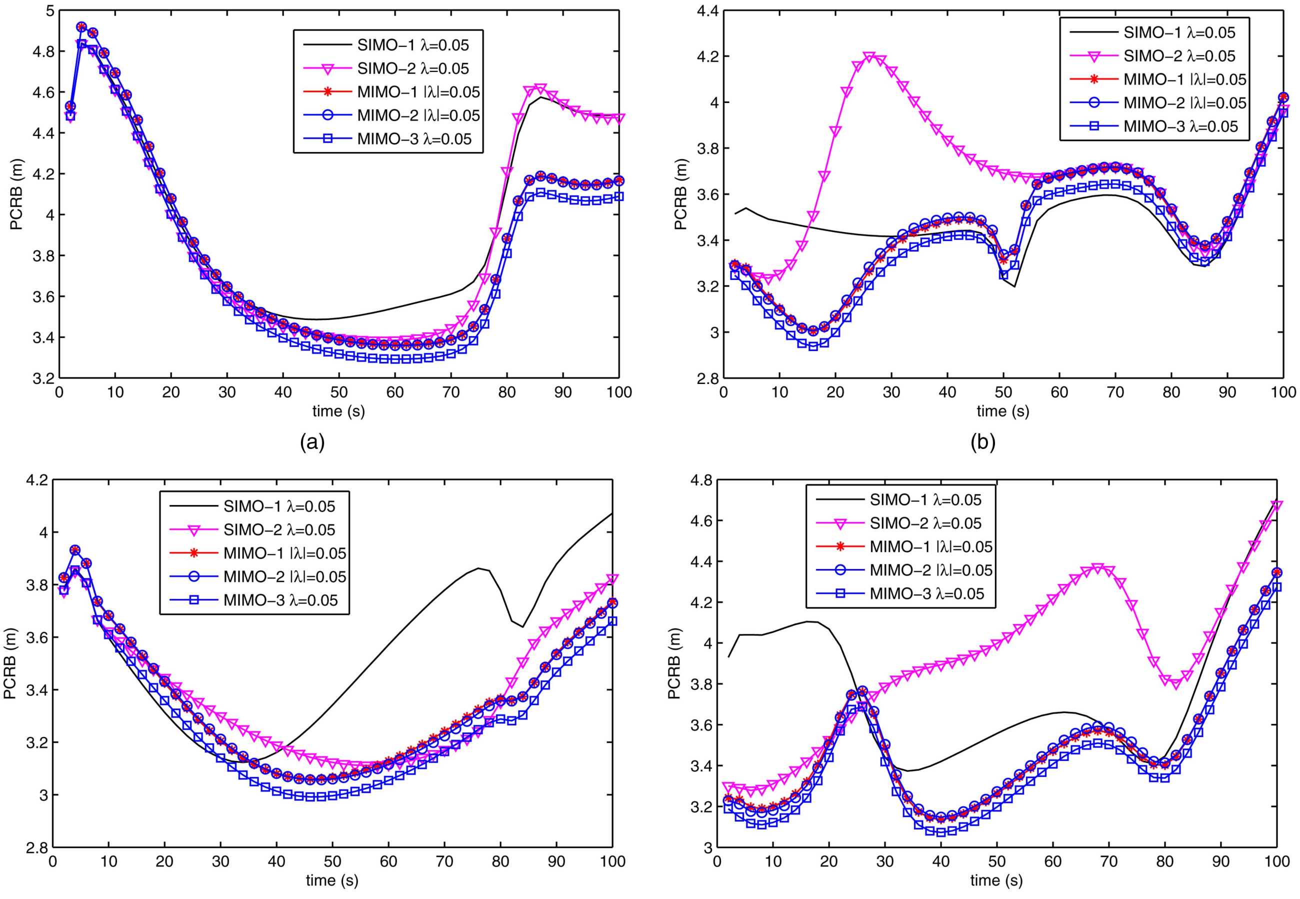
based on (54). Since $\mathrm{Diag}([0,\lambda^2]) \succeq \mathbf{0}$ and $\mathbf{G} \succeq 0$, we have $\mathbb{E}\{\bar{\mathbf{P}}_{k+1}^{\mathrm{ML}}(\lambda)\} \succeq \mathbb{E}\{\bar{\mathbf{P}}_{k+1}^{\mathrm{SL}}(0)\}$. Due to the fact that $\mathbf{J}_k^{\mathrm{ML}}(\lambda,2\sigma_r^2) = \mathbf{J}_k^{\mathrm{SL}}(0,\sigma_r^2)$, *Lemma 3* can be proven.

To sum up, a co-located MIMO configuration with waveform parameters $\pm\lambda$ can be better than a zero λ multistatic radar, but worse than the multistatic radar with waveform parameter $|\lambda|$ in a power limited scenario. Numerical simulations follow to demonstrate these conclusions. The MIMO radar configuration is the same as the multistatic one in Section 6. The range variance is $2\sigma_r^2$ for the former but σ_r^2 for the latter; the other parameters of MIMO radar keep the same as the multistatic ones. The MIMO PCRBs are in Fig. 7 for different λs and target trajectories. Combining Figs. 4 and 7, it is obvious that the MIMO PCRBs are below that for the multistatic one with $\lambda = 0$, but above those corresponding with parameters $|\lambda|$.

Note that the proof of *Lemma 3* assumes that the trajectory is far away from the antennas; however, the simulation, for example Fig. 7, shows that the conclusion still holds even without the assumption.

2) *Transmitter Widely-separated Case*: If the two transmitters are widely-separated, geometric diversity is improved. The main concern here is whether a MIMO configuration is always better than a multistatic one under a constraint on energy. Based on Section 6, a multistatic radar with positive λ is better than that with negative or zero ones, so the multistatic case with $\lambda \le 0$ is not considered in comparison.

We inherit the multistatic constellation and trajectories in Fig. 3, and add another transmitter, say T-2, at (20 km, 20 km). The two transmitters evenly share the total energy. Based the previous section, a multistatic radar with a positive λ seems to be best, so we only consider a multistatic radar with a positive λ in this part. The results are in Fig. 8, where 'SIMO-1' and 'SIMO-2' denote the PCRBs for the multistatic radars respectively with T-1 and T-2 as transmitter. 'MIMO-1' represents the scenario with transmitting strategies $\{\text{T-1}, |\lambda|\}$ and



(a)

(b)

(c)

(d)

Fig. 8. PCRBs of a transmitter widely-separated MIMO radar with different waveform parameters λs and target trajectories. The prior distribution is $P_0 = \mathrm{diag}([20,20,5,5])$.

{T-2,$-|\lambda|$}, and 'MIMO-2' is for strategies {T-1,-$|\lambda|$} and {T-2,$|\lambda|$}. Finally, 'MIMO-3' stands for the case where both transmitters employ up-sweep LFM; such a system is only possible if the two carrier frequencies are sufficiently separated so as to avoid spectrum interference. From those figures, we observe:

- The PCRBs of 'MIMO-3' are uniformly better than those of 'MIMO-1' and 'MIMO-2' for different trajectories. From a perspective of tracking accuracy, 'MIMO-3' is the best among these three scenarios, even though it has a low spectrum efficiency.
- The PCRBs are highly geometry dependent. As for 'MIMO-3,' 'SIMO-1,' and 'SIMO-2,' no strategy is uniformly better than others.

In short, no uniformly best strategy exists. A clever MIMO radar could adjust transmission scenarios, including waveform assignments and transmitter on-off controlling, to improve its performance.

8. CONCLUSIONS

Range-Doppler coupling is an important characteristic for a Doppler tolerant waveform, where the extracted range is biased by the unknown Doppler shift. The tracking performance of distributed radar systems with range measurements was investigated with the help of the PCRB, and the PCRBs for various parameters and antenna configurations were compared under different energy constraints. The detection processes are assumed ideal, no missed detections nor false alarms. In brief, a waveform with positive range-Doppler coupling (that is, a negative range/range-rate coupling) is a good choice for a multistatic radar or an energy constrained colocated MIMO radar. However, for a transmitter widely-separated configuration, the PCRB is extremely geometry dependent; no uniformly best scenario exists under a power constraint.

REFERENCES

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation with Application to Tracking and Navigation.
Wiley-Interscience, 2001.

[2] R. J. Fitzgerald
Effects of range-Doppler coupling on Chirp radar tracking accuracy.
IEEE Transactions on Aerospace and Electronic Systems, **10**, 4 (July 1974), 528–532.

[3] J. E. Gentle
Elements of Computational Statistics.
Springer, 2002.

[4] A. M. Haimovich, R. S. Blum, and L. J. Cimini
MIMO radar with widely separated antennas.
IEEE Signal Processing Magazine, **25**, 1 (Jan. 2008), 116–129.

[5] E. J. Kelly and R. P. Wishner
Matched-filter theory for high-velocity, accelerating targets.
IEEE Transactions on Military Electronics, **9**, 1 (Jan. 1965), 56–69.

[6] N. Levanon and E. Mozeson
Radar Signals.
John Wiley & Sons, Inc., Hoboken, NJ, 2004.

[7] J. Li and P. Stoica
MIMO radar with colocated antennas.
IEEE Signal Processing Magazine, **24**, 5 (Sept. 2007), 106–114.

[8] R. Niu, P. Willett, and Y. Bar-Shalom
Matrix CRLB scaling due to measurements of uncertain origin.
IEEE Transactions on Signal Processing, **49**, 7 (July 2001), 1325–1335.

[9] R. Niu, P. Willett, and Y. Bar-Shalom
Tracking considerations in selection of radar waveform for range and range-rate measurements.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 2 (Apr. 2002), 467–487.

[10] X. Song, P. Willett, and S. Zhou
Posterior Cramér-Rao bounds for Doppler biased multistatic range-only tracking.
In *Proceedings of the International Conference on Information Fusion*, Chicago, IL, July 2011.

[11] X. Song, S. Zhou, and P. Willett
Reducing the waveform cross correlation of MIMO radar with space-time coding.
IEEE Transactions on Signal Processing, **58**, 8 (Aug. 2010), 4213–4224.

[12] P. Tichavsky, C. H. Muravchik, and A. Nehorai
Posterior Cramér-Rao bounds for discrete-time nonlinear filtering.
IEEE Transactions on Signal Processing, **46**, 5 (May 1998), 1386–1396.

[13] H. Van Trees
Detection, Estimation, and Modulation Theory.
John Wiley & Sons, Inc., NY, 1 ed., 1968.

[14] W. Wong and W. D. Blair
Steady-state tracking with LFM waveforms.
IEEE Transactions on Aerospace and Electronic Systems, **36**, 2 (Apr. 2000), 701–709.

[15] X. Zhang, P. Willett, and Y. Bar-Shalom
Tracking considerations in selection of radar waveform given range measurements.
In *Proceedings of 2001 SPIE Annual Conference on Signal and Data Processing of Small Targets*, San Diego, CA, Aug. 2001, vol. 4473, pp. 130–141.

[16] X. Zhang, P. Willett, and Y. Bar-Shalom
Dynamic Cramér-Rao bound for target tracking in clutter.
IEEE Transactions on Aerospace and Electronic Systems, **41**, 4 (Oct. 2005), 1154–1167.

[17] Y. Zhang, G. J. Frazer, and M. G. Amin
Concurrent operation of two over-the-horizon radars.
IEEE Journal of Selected Topics in Signal Processing, **1**, 1 (June 2007), 114–123.

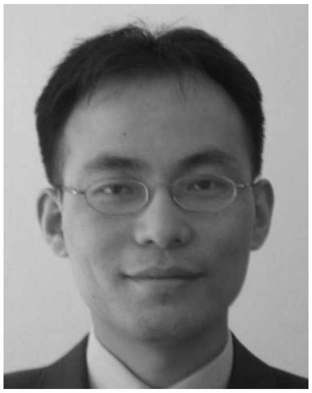
Xiufeng Song (S’08) received the B.S. degree from Xidian University, Xi’an, China, in 2005 and the M.S. degree from Institute of Electronics, Chinese Academy of Sciences (CAS), Beijing, China, in 2008, both in electrical engineering.

He is currently working towards the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs. His research interests lie in signal processing, detection, and estimation theory.

Peter Willett (F'03) received his B.A.Sc. (engineering science) from the University of Toronto in 1982, and his Ph.D. degree from Princeton University in 1986.

He has been a faculty member at the University of Connecticut since 1986, and since 1998 has been a professor. He has published 135 journal articles (13 more under review), 290 conference papers, and 9 book chapters. His primary areas of research have been statistical signal processing, detection, machine learning, data fusion and tracking. He has interests in and has published in the areas of change/abnormality detection, optical pattern recognition, communications and industrial/security condition monitoring.

He is editor-in-chief for *IEEE Transactions on Aerospace and Electronic Systems*, and until recently was associate editor for three active journals—*IEEE Transactions on Aerospace and Electronic Systems* (for Data Fusion and Target Tracking) and *IEEE Transactions on Systems, Man, and Cybernetics*, parts A and B. He is also associate editor for the IEEE AES Magazine, editor of the AES Magazine's periodic Tutorial issues, associate editor for ISIF's electronic *Journal of Advances in Information Fusion*, and is a member of the editorial board of IEEE's Signal Processing Magazine. He was a member of the IEEE AESS Board of Governors 2003–2009. He was general cochair (with Stefano Coraluppi) for the 2006 ISIF/IEEE Fusion Conference in Florence, Italy, Program Co-Chair (with Eugene Santos) for the 2003 IEEE Conference on Systems, Man & Cybernetics in Washington, D.C., and program cochair (with Pramod Varshney) for the 1999 Fusion Conference in Sunnyvale. He was coorganizer of the tracking subsession at the 1999 IEEE Aerospace Conference, and has been organizer of the Remote Sensing Track of that conference 2000–2003. Jointly with T. Kirubarajan he has coorganized the SPIE "System Diagnosis and Prognosis: Security and Condition Monitoring Issues" Conference in Orlando, 2001–2003. He has been a member of the IEEE Signal Processing Society's Sensor-Array & Multichannel (SAM) Technical Committee since 1997, and both serves on that TC's SAM Conferences' Program Committees and maintains the SAM website.



Shengli Zhou (SM'11) received the B.S. degree in 1995 and the M.Sc. degree in 1998, from the University of Science and Technology of China (USTC), Hefei, both in electrical engineering and information science. He received his Ph.D. degree in electrical engineering from the University of Minnesota (UMN), Minneapolis, in 2002.

He has been an assistant professor with the Department of Electrical and Computer Engineering at the University of Connecticut (UCONN), Storrs, 2003–2009, and now is an associate professor. He holds a United Technologies Corporation (UTC) Professorship in Engineering Innovation, 2008–2011. His general research interests lie in the areas of wireless communications and signal processing. His recent focus is on underwater acoustic communications and networking.

Dr. Zhou served as an associate editor for *IEEE Transactions on Wireless Communications*, Feb. 2005–Jan. 2007, and *IEEE Transactions on Signal Processing*, Oct. 2008–Oct. 2010. He is now an associate editor for *IEEE Journal of Oceanic Engineering*. He received the 2007 ONR Young Investigator award and the 2007 Presidential Early Career Award for Scientists and Engineers (PECASE).

Performance Prediction of Multisensor Tracking Systems for Single Maneuvering Targets

WILLIAM D. BLAIR
PAUL A. MICELI

Studying the performance of multisensor tracking systems against maneuvering targets involves Monte Carlo simulations with the tracking algorithms implemented in a sophisticated computer simulation of the multisensor system. However, a simplified method for predicting the performance of a multisensor tracking system against maneuvering targets is needed for confirmation of the computer simulations, real-time command and control decisions such as multisensor resource allocation, and systems engineering of complex multisensor systems. The challenge of accurate performance prediction arises from the lack of covariance consistency of the Kalman filter when tracking maneuvering targets. In this paper, a method for performance prediction of a nearly constant velocity Kalman filter is extended to tracking a maneuvering target with multiple dispersed sensors on an oblate earth. Given target position and acceleration as a function of time, the tracking performance of each sensor is expressed as a sensor-noise only (SNO) covariance and maneuver lag or filter bias. In the fusion of the data from the multiple sensors, the SNO covariances fuse for a smaller covariance, while the maneuver lags fuse with a gain proportional to the inverse of the covariances for the sensor tracks. This method can also be used to predict the performance of a multisensor system that include one, two, and/or three dimensional sensors. The results of Monte Carlo simulations of multisensor tracking of a maneuvering target are used to illustrate the accuracy of methods for performance prediction.

Manuscript received May 13, 2008; revised April 13, 2009 and July 18, 2011; released for publication July 26, 2011 .

Refereeing of this contribution was handled by Yaakov Bar-Shalom.

Authors' address: Georgia Tech Research Institute, Georgia Institute of Technology, Atlanta, Georgia 30332-8157, E-mail: (dale.blair@gtri.gatech.edu).

1557-6418/12/$17.00 © 2012 JAIF

1. INTRODUCTION

Studying the performance of multisensor tracking systems against maneuvering targets involves Monte Carlo simulations with the tracking algorithms implemented in a sophisticated computer simulation of the multisensor system [2], [1]. However, a simplified method for predicting the performance of a multisensor tracking system against maneuvering targets is needed to confirm the results of computer simulations, real-time command and control decisions such as multisensor resource allocation, and engineering of complex multisensor systems [3].

When reviewing current approaches to performance prediction, four traits are helpful in distinguishing the relative advantages between the approaches listed in the literature survey and the work presented in this paper. The first two traits deal with the ability to predict performance of multisensor or multitarget tracking algorithms. Most of the current approaches to performance prediction typically deal with a single sensor and single target. In addition, while some algorithms deal with either the multisensor or multitarget case, no work has been identified that is capable of handling both the performance prediction of multisensor and multitarget tracking algorithms. The third trait is ability to predict performance of a defined scenario. While some performance prediction techniques express an expected performance based on sensor parameters, none extend well to maneuvering target scenarios since deterministic changes in motion are not considered. Finally, the fourth trait to be considered is algorithm complexity. Engineering of complex systems often involves extensive parametric variability for which, traits three and four are critical. Algorithm scenario dependence and simplicity yield a high level of confidence when comparing against more complicated simulated algorithms. In this work, we present a simple to compute algorithm that accounts for scenario laydowns and can be extended to multiple sensors while assuming measurement sharing on a single maneuvering target.

Cramer-Rao Bound (CRB) techniques are one of the basic tools for estimating performance. However, these techniques were originally designed for estimating deterministic parameters. More appropriate for the performance prediction of tracking systems, the Posterior Cramer-Rao Bound (PCRB) extends the CRB to provide a "measure" of system performance when both measurements and state are assumed to be stochastic processes [5], [4]. The work of [6] extends the PCRB to handle the prediction of multitarget systems under a set of assumptions. In addition, the work of [7] extends the PCRB to handle the prediction of an estimator for a single maneuvering target. However, none of the papers dedicated to an extension of the PCRB include comparisons to Monte Carlo simulations from a realistic tracker. Thus, the usefulness of each algorithm as it pertains to the performance of a multitarget tracker in a given scenario is not known.

A particular method of performance prediction that compares well to Monte Carlo simulations is the hybrid conditional averaging (HYCA) technique [8]. The unique feature of this algorithm is that it allows for the performance prediction of algorithms where the uncertainties involved are both continuous and discrete. Such is the case for the interacting multiple model (IMM) algorithm, where a HYCA technique specifically tailored for performance prediction of an IMM estimator [9]. While methods based on HYCA compare well to Monte Carlo simulations, the calculations are quite complex. In addition, the existing methods typically only apply to single sensor and single target scenarios.

One extension of the HYCA technique to multisensor multitarget applications is given in [10] where, the Joint Probabilistic Data Association (JPDA) algorithm is of interest. Specifically, two implementations of the algorithm, sequential and parallel, are compared through the use of performance prediction. Though the JPDA algorithm is specifically tailored for multitarget applications, the performance prediction technique in [10] was not extended to multiple targets. Finally, the work in [11] can be extended to multiple sensors but not for multiple targets [12].

Most of the current research in performance prediction with respect to target tracking does not take into account multisensor or multi-target tracking scenarios. PCRB methods do not take into account specific scenarios (they average over an ensemble), and therefore cannot be compared to Monte Carlo simulations. Some techniques such as HYCA compare very well to Monte Carlo simulations, but the calculations are quite complex and have not been extended to multiple targets. In this paper, a multisensor performance prediction technique for a single maneuvering target that is easy to compute and compares very well with Monte Carlo simulations is developed. Therefore, even though the multitarget case is currently not considered, the method presented in this paper has unique advantages in the aggregate when compared to other methods of performance prediction. Future work will include extensions to multisensor and multitarget scenarios.

The challenge of accurate performance prediction arises from the lack of covariance consistency of the Kalman filter when tracking maneuvering targets. For single sensor tracking of maneuvering targets, the lack of covariance consistency is addressed through the use of steady-state filter analysis to decompose the performance characterization into a sensor noise only (SNO) covariance and maneuver lag or filter bias [15], [20], [16]. Thus, given the sensor location, the position and acceleration as a function of time, the parameters for the sensor measurement errors, and the process noise covariance assumed by the filter, the root mean square error (RMSE) in both position and velocity as a function of time can be predicted for a nearly constant velocity Kalman filter. Since the typical variances of the measurement errors in each coordinate are constant or vary slowly, and an alpha-beta filter characterizes Kalman filtering under steady state conditions, the alpha beta filter equations can be used to approximate the tracking performance of each coordinate of the sensor. The SNO covariance and the maneuver lags are generated in each of the sensor coordinates separately as the measurement errors tend to be independent between coordinates [16]. The maneuver lag represents the component of the state estimate error due to a deterministic target maneuver and is assumed constant for a certain time [20]. The SNO covariance represents the stochastic part of the state estimate error due to measurement noise. For each coordinate, typical values of the measurement variance in the corresponding Cartesian coordinate, measurement rate, and the process noise variance of the track filter are used to select the filter gains for an alpha-beta filter, and the SNO covariance and maneuver lag are computed from the gains. Given the SNO covariances and the maneuver lags for all three coordinates, the full track covariance matrix and bias vector can be generated in sensor coordinates, and those can be transformed to any coordinate system.

This method for performance prediction can be extended to nearly constant velocity track filtering with multiple dispersed sensors on an oblate earth. A similar method was utilized in [17] without a rigorous presentation. As in the case of single sensor tracking, the challenge of accurate performance prediction arises from the lack of covariance consistency in the nearly constant velocity Kalman filter. However, multiple sensor frames exist in this case and the technique in [17] must be extended to address this difference. This paper presents a rigorous and detailed development of a method for predicting the performance of nearly constant velocity tracking of a single maneuvering target with multiple sensors in the absence of data association errors. Thus, the predicted performance represents a lower bound for the track errors for nearly constant velocity tracking. The computed SNO covariance and maneuver lags for each sensor used in the equations for the fusion of multiple independent tracks to characterize the fused SNO covariance and maneuver lag. The prediction methods implemented by these equations constitute the significant contribution of this work. In the fusion of the multisensor data, the SNO covariances fuse for a smaller covariance, while the maneuver lags fuse with a gain proportional to the inverse of the covariances for the bias in the sensor tracks. This method can also be used to predict the performance of a multisensor system that include one, two, and/or three dimensional sensors as long as a combination of the sensors provide observability of all three coordinates. The predicted performance represents that of a multisensor tracking system that shares associated measurement reports between sensors and uses nearly constant velocity Kalman filtering for state estimation.

This paper is organized as follows. Background information is given in Section 2 to familiarize the reader with the notation and techniques that serve as the foundation for the methods developed. Least squares estimation, Kalman filtering, alpha-beta filtering, performance prediction of alpha-beta filters, and track fusion are reviewed briefly. In Section 3, Monte Carlo simulation results involving the nearly constant velocity tracking of a maneuvering target are given to illustrate the performance prediction for a single sensor tracking a maneuvering target. In Section 4, the techniques are expanded to the performance prediction of multisensor tracking of a single maneuvering target. The results of Monte Carlo simulations for multisensor tracking of a maneuvering target are shown in Section 5 to verify the methods for performance prediction. Concluding remarks are given in Section 6.

2. BACKGROUND

2.1. Least Squares Estimation

Linear least squares estimation is a mathematical technique for parameter estimation that attempts to find the best linear fit to a set of data, where the best is the parameter value that minimizes the sum of the errors squared. The parameters are assumed to be unknown and time-invariant. Consider the linear observation modeled as

$$Y_j = H_j X + W_j \tag{1}$$

where Y_j is the jth observation j, H_j is known for observation j, X is fixed and unknown, and $W_j \sim (0, R_j)$ is the error in the jth observation. For the expression in (1), an estimate of X may be obtained by minimizing the cost function for a least squares estimator (LSE) that is defined for independent observations as

$$C(X) = (Y - HX)^T R^{-1} (Y - HX)$$

where

$$H = [H_1 \quad H_2 \cdots H_N]^T \tag{2}$$

$$Y = [Y_1 \quad Y_2 \cdots Y_N]^T \tag{3}$$

$$W = [W_1 \quad W_2 \cdots W_N]^T \tag{4}$$

$$R = E[WW^T] = \begin{bmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & R_N \end{bmatrix}. \tag{5}$$

The cost is minimized by taking the partial derivative with respect X and setting the result equal to zero. Rearranging terms to solve for the estimate of X gives

$$\hat{X} = (H^T R^{-1} H)^{-1} H^T R^{-1} Y \tag{6}$$

and the covariance of $\hat{X}$ is given by [21]

$$\mathrm{COV}(\hat{X}) = (H^T R^{-1} H)^{-1}. \tag{7}$$

For Gaussian errors W_j, $\hat{X}$ is the maximum likelihood and minimum mean squared error (MMSE) estimate of X.

2.2. Kalman Filter

A Kalman filter is often employed to filter the kinematic measurements for estimating the position, velocity, and acceleration of a target [1]. The kinematic model commonly assumed for a target in track is given by

$$X_{k+1} = F_k X_k + G_k \nu_k \tag{8}$$

where $\nu_k \sim N(0, Q_k)$ is the process noise that models the unknown target acceleration and F_k defines the linear dynamics. The target state vector X_k contains the position, velocity, and possibly acceleration of the target at time t_k, as well as other variables used to model the time-varying acceleration. For this paper, X_k will include position and velocity. The linear measurement model is given by

$$Y_k = H_k X_k + w_k \tag{9}$$

where Y_k is typically the measurement of the position of the target and $w_k \sim N(0, R_k)$ is the observation error. Both w_k and ν_k are assumed to be independent "white" noise processes. When designing the Kalman filter, Q_k is selected such that the 65% to 95% confidence region about zero contains the maximum acceleration level of the target. However, when targets maneuver, the acceleration changes in a deterministic manner. Thus, the white noise assumption associated with ν_k is often violated and the filter develops a bias in the state estimates. If a larger Q_k is chosen, the bias in the state estimates is less during a maneuver, but then Q_k characterizes poorly the target motion when the target is not maneuvering and the filter performance is far from optimal. Furthermore, the error in modeling the two modes of motion (i.e., nonmaneuvering and maneuvering) with a single model and the error in the white noise assumption for the process noise during maneuvers result in an inaccurate state error covariance that cannot be used reliably for performance prediction. While an Interacting Multiple Model (IMM) estimator [1] can be used to address this conflict, the primary focus of this work is on the quick prediction of average tracking performance and that performance serves as an optimistic guide for track filtering performance of targets maneuvering in the presence of data association errors. Given the mode switching of an IMM estimator is not considered in this work, the performance prediction algorithms presented here can be viewed as an estimate of the upper bound of the expected errors for an IMM estimator.

2.3. Alpha-Beta Filter

The alpha-beta filter is based on the assumption that the target is moving with constant velocity plus zero-mean, white Gaussian acceleration errors. In order to simplify the example and permit analytical predictions of the filter performance, the motion of the target is defined in a single coordinate and the measurements are the positions of the target (i.e., a linear function of the state). For the alpha-beta filter and nearly constant velocity Kalman filter, the state and measurement equations of (8) and (9) are defined by

$$X_k = [x_k \quad \dot{x}_k]^T \tag{10}$$

$$F_k = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^T \tag{11}$$

$$G_k = \begin{bmatrix} \dfrac{1}{2\sqrt{3}} T^{3/2} & \dfrac{1}{2} T^{3/2} \\ 0 & T^{1/2} \end{bmatrix} \tag{12}$$

$$H_k = [1 \quad 0] \tag{13}$$

where $R_k = \sigma_w^2$ is the variance of the measurement errors in m^2, T is the time interval between measurements, $Q_k = \sigma_v^2 I_{2\times 2}$ is the process noise covariance matrix with $\sigma_v^2 T^{-1}$ denoting the variance of the "acceleration" errors in $\mathrm{m}^2/\mathrm{s}^4$, and σ_v^2 denotes the power spectral density (PSD). This process noise model corresponds to the continuous white noise acceleration (CWNA) model [1].

The steady-state form of the constant velocity filter is used for analytical predictions of filter performance. For a filter to achieve these steady-state conditions, the error processes ν_k and w_k must be stationary and the data rate must be constant. While these conditions are seldom satisfied in practice, the steady-state form of the filter can be used to predict average or expected tracking performance. The alpha-beta filter is equivalent to the Kalman filter in steady-state for this motion model. For the alpha-beta filter, the steady-state gains that occur after the transients associated with filter initialization diminish are given by

$$K_k = \begin{bmatrix} \alpha & \dfrac{\beta}{T} \end{bmatrix}^T \tag{14}$$

where α and β are the optimal gains for the CWNA model. As given in [1], the gains are computed by

$$\Gamma^2 = \frac{\sigma_v^2 T^3}{\sigma_w^2} \tag{15}$$

$$\mu = \frac{1}{3} + \sqrt{\frac{1}{12} + \frac{4}{\Gamma^2}} \tag{16}$$

$$\alpha = \beta\sqrt{\mu} \tag{17}$$

$$\beta = \frac{12}{6\left(\mu + \sqrt{\mu}\right) + 1} \tag{18}$$

where Γ is known as the tracking maneuver index. The steady-state error covariance of the alpha-beta filter [1], [13] is given by

$$P_{k|k}^{\alpha\beta} = \sigma_w^2 \begin{bmatrix} \alpha & \dfrac{\beta}{T} \\ \dfrac{\beta}{T} & \dfrac{\beta(2\alpha-\beta)}{2(1-\alpha)T^2} \end{bmatrix}. \tag{19}$$

A simple gain scheduling procedure for approximating the performance of a Kalman filter during initialization from [13] is given by ($k = 0$ for the first measurement)

$$\alpha_k = \max\left\{ \frac{2(2k+1)}{(k+1)(k+2)}, \alpha \right\} \tag{20}$$

$$\beta_k = \max\left\{ \frac{6}{(k+1)(k+2)}, \beta \right\} \tag{21}$$

where α and β are the steady-state values and the initial conditions are given by

$$X_{0|-1} = \begin{bmatrix} x_{0|-1} \\ \dot{x}_{0|-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{22}$$

2.4. Performance Prediction for Single Sensor Tracking of a Maneuvering Target

The covariance of the state estimate $X_{k|k}$ is given by

$$P_{k|k} = E[(X_{k|k} - \bar{X}_{k|k})(X_{k|k} - \bar{X}_{k|k})^T \mid X_k] \tag{23}$$

where $E[\cdot]$ denotes the expected value operator, and $\bar{X}_{k|k} = E[X_{k|k}]$. When the estimator is unbiased and $E[X_{k|k}] = X_k$, the true value, the covariance is a good predictor of performance. However, when the estimator is biased, the covariance is a poor predictor of performance. When a target undergoes a deterministic maneuver (i.e., a constant acceleration), the estimates are biased and the covariance matrix tends to be an optimistic estimate of track filter performance since it does not reflect the bias. When a target undergoes no maneuver (i.e., a zero acceleration), the covariance matrix tends to also be a biased estimate of track filter performance, because process noise is included in the filter for maneuver response. Thus, in order to address both conditions of the performance prediction, the mean-squared error will be written in terms of a SNO covariance for no maneuver and a bias or maneuver lag for the constant acceleration maneuver.

Let

$$B_{k|k} = E[X_{k|k} \mid X_k] - X_k = \bar{X}_{k|k} - X_k \tag{24}$$

where $B_{k|k}$ denotes the filter bias. Thus, mean squared error (MSE) is given by

$$\begin{aligned} \mathrm{MSE} &= E[(X_{k|k} - X_k)(X_{k|k} - X_k)^T \mid X_k] \\ &= E[(X_{k|k} - \bar{X}_{k|k})(X_{k|k} - \bar{X}_{k|k})^T \\ &\qquad + 2(X_{k|k} - \bar{X}_{k|k})(\bar{X}_{k|k} - X_k)^T \\ &\qquad + (\bar{X}_{k|k} - X_k)(\bar{X}_{k|k} - X_k)^T \mid X_k] \\ &= P_{k|k} + B_{k|k}B_{k|k}^T. \end{aligned} \tag{25}$$

Consider the case of deterministic maneuvers of either zero acceleration or constant acceleration, the filter covariance is given by the SNO covariance when the acceleration is zero and the SNO covariance plus the bias error squared when the acceleration is a nonzero constant. Letting $S^{\alpha\beta}_{k|k}$ denote the SNO covariance of the alpha-beta filter and $B^{\alpha\beta}_{k|k}$ denote the bias due to an acceleration gives

$$\begin{aligned} &E[(X_{k|k} - X_k)(X_{k|k} - X_k)^T \mid X_k] \\ &\quad = S^{\alpha\beta}_{k|k} + B^{\alpha\beta}_{k|k}(B^{\alpha\beta}_{k|k})^T. \end{aligned} \tag{26}$$

The SNO covariance and the bias are computed by representing the alpha-beta filter as a linear, time-invariant system with an input that can be expressed as a deterministic signal (i.e., a constant acceleration rather than zero-mean white process noise) with white noise measurement errors. The input-output relationships between the measurements Y_k and state estimate $X_{k|k}$ can be expressed as a linear system that is given by

$$X_{k|k} = \bar{F}_k X_{k-1|k-1} + \bar{G}_k Y_k$$

where

$$\bar{F}_k = \begin{bmatrix} 1-\alpha & (1-\alpha)T \\ -\dfrac{\beta}{T} & 1-\beta \end{bmatrix} \tag{27}$$

$$\bar{G}_k = \begin{bmatrix} \alpha & \dfrac{\beta}{T} \end{bmatrix}^T. \tag{28}$$

The error covariance of $X_{k|k}$ that results from the SNO (i.e., no target acceleration to produce bias) is given in [13], [14], and [18] for arbitrary α and β to be

$$S^{\alpha\beta}_{k|k} = \frac{\sigma_w^2}{\alpha(4-2\alpha-\beta)} \begin{bmatrix} 2\alpha^2 + \beta(2-3\alpha) & \dfrac{\beta}{T}(2\alpha-\beta) \\ \dfrac{\beta}{T}(2\alpha-\beta) & \dfrac{2\beta^2}{T^2} \end{bmatrix} \tag{29}$$

where T is the time period between consecutive measurements and σ_w^2 is the variance of measurement errors. Since (29) includes only the sensor measurement errors, it is referred to as the SNO covariance matrix. For a maneuvering target, the bias or lag in the state estimate for arbitrary α and β is given by

$$B^{\alpha\beta}_{k|k} = \begin{bmatrix} (1-\alpha)\dfrac{T^2}{\beta} \\ \left(\dfrac{\alpha}{\beta} - 0.5\right)T \end{bmatrix} A_k \tag{30}$$

where A_k is the acceleration of the target in the coordinate of interest at time t_k. This is the steady-state bias that results after the transient response of the filter has decayed (i.e., typically three or four measurements) [20].

For an m-step (i.e., measurement intervals) ahead prediction, the error covariance of the state estimate that results from the measurement errors only is given in [17] for an arbitrary α and β to be

$$S^{\alpha\beta}_{k+m|k} = F(m) S^{\alpha\beta}_{k|k} F(m)^T \tag{31}$$

where

$$F(m) = \begin{bmatrix} 1 & mT \\ 0 & 1 \end{bmatrix}. \tag{32}$$

Thus, for an m-step ahead prediction, the SNO covariance is given by

$$S^{\alpha\beta}_{k+m|k} = \frac{\sigma_w^2}{\alpha(4-2\alpha-\beta)} \begin{bmatrix} E_{11} & E_{12} \\ E_{12} & E_{22} \end{bmatrix} \tag{33}$$

where

$$\begin{aligned} E_{11} &= 2\alpha^2 + \beta(2-3\alpha) + 2m\beta(2\alpha-\beta) + 2m^2\beta^2 \\ E_{12} &= \frac{\beta}{T}(2\alpha - \beta + 2m\beta) \\ E_{22} &= \frac{2\beta^2}{T^2}. \end{aligned}$$

For an m-step ahead prediction and a maneuvering target, the bias or lag in the state estimate for arbitrary α and β is given by [17] and [18] as

$$B^{\alpha\beta}_{k+m|k} = \begin{bmatrix} (1-\alpha+(\alpha-0.5\beta)m + 0.5\beta m^2)\dfrac{T^2}{\beta} \\ (\alpha + (m-0.5)\beta)\dfrac{T}{\beta} \end{bmatrix} A_k. \tag{34}$$

The RMSE in the position estimates of the alpha-beta filter for an m-step ahead prediction is given by

$$\begin{aligned} &\mathrm{RMSE}^p(m) \\ &= \left[\frac{\sigma_w^2}{\alpha(4-2\alpha-\beta)}(2\alpha^2 + \beta(2-3\alpha) + 2m\beta(2\alpha-\beta) + 2m^2\beta^2) \right. \\ &\quad \left. + (1-\alpha+(\alpha-0.5\beta)m + 0.5\beta m^2)^2 \frac{T^4}{\beta^2} A_k^2 \right]^{1/2}. \end{aligned} \tag{35}$$

The RMSE in the velocity estimates can be expressed as

$$\begin{aligned} &\mathrm{RMSE}^v(m) \\ &= \left[\frac{2\sigma_w^2\beta^2}{\alpha(4-2\alpha-\beta)T^2} + ((\alpha + (m-0.5)\beta)\frac{T}{\beta})^2 A_k^2\right]^{1/2}. \end{aligned} \tag{36}$$

Note that while m in (31) through (36) is treated as an integer, the results are valid for a fractional measurement interval as well.

2.5. Track Fusion for N Independent Tracks

The fusion of N uncorrelated or independent tracks can be formulated as a linear least-squares estimation problem, where the sensor tracks are treated as observations with independent errors. For N independent tracks with mean and covariance $\{X^i_{k|k}, P^i_{k|k}\}^N_{i=1}$,

$$H = [I_N \quad I_N \cdots I_N]^T \tag{37}$$

$$Y = [X^1_{k|k} \quad X^2_{k|k} \cdots X^N_{k|k}]^T \tag{38}$$

$$R = \begin{bmatrix} P_{k|k}^1 & 0 & \cdots & 0 \\ 0 & P_{k|k}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & P_{k|k}^N \end{bmatrix}. \tag{39}$$

The block diagonal form allows for the inversion of each block individually. Thus, the fused track and covariance is given by

$$X_{k|k} = \left[\sum_{i=1}^{N} (P_{k|k}^i)^{-1}\right]^{-1} \left[\sum_{i=1}^{N} (P_{k|k}^i)^{-1} X_{k|k}^i\right] \tag{40}$$

$$P_{k|k} = \left[\sum_{i=1}^{N} (P_{k|k}^i)^{-1}\right]^{-1}. \tag{41}$$

2.6. Track Fusion for Two Correlated Tracks

For two correlated tracks, the errors in the estimator are correlated and the covariance is represented as

$$R_k = E[W_k W_k^T] = \begin{bmatrix} P_{k|k}^1 & P_{k|k}^{12} \\ (P_{k|k}^{12})^T & P_{k|k}^2 \end{bmatrix}. \tag{42}$$

The inverse of this matrix is given by

$$R_k^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{43}$$

where

$$A_{11} = [P_{k|k}^1 - P_{k|k}^{12}(P_{k|k}^2)^{-1}(P_{k|k}^{12})^T]^{-1}$$
$$A_{12} = -(P_{k|k}^1)^{-1}P_{k|k}^{12} \times [P_{k|k}^2 - (P_{k|k}^{12})^T(P_{k|k}^1)^{-1}P_{k|k}^{12}]^{-1}$$
$$A_{21} = -(P_{k|k}^2)^{-1}(P_{k|k}^{12})^T \times [P_{k|k}^1 - P_{k|k}^{12}(P_{k|k}^2)^{-1}(P_{k|k}^{12})^T]^{-1}$$
$$A_{22} = [P_{k|k}^2 - (P_{k|k}^{12})^T(P_{k|k}^1)^{-1}P_{k|k}^{12}]^{-1}.$$

Thus, the fused track is given by

$$X_{k|k} = [D_1 + D_2]^{-1}\{D_1 X_{k|k}^1 + D_2 X_{k|k}^2\} \tag{44}$$

where

$$D_1 = [I - (P_{k|k}^2)^{-1}(P_{k|k}^{12})^T] \times [P_{k|k}^1 - P_{k|k}^{12}(P_{k|k}^2)^{-1}(P_{k|k}^{12})^T]^{-1}$$
$$D_2 = [I - (P_{k|k}^1)^{-1}P_{k|k}^{12}] \times [P_{k|k}^2 - (P_{k|k}^{12})^T(P_{k|k}^1)^{-1}P_{k|k}^{12}]^{-1}.$$

Thus, the fusion of correlated tracks can be accomplished. However, analytical expressions are not easily achieved for more than two tracks. Furthermore, calculation of the track correlation further complicates the performance prediction process. For the case of measurement level fusion, the performance prediction is more appropriately matched by ignoring the correlation.

3. PREDICTING PERFORMANCE FOR A SINGLE SENSOR

As a numerical example, a radar tracking system with a 1 Hz measurement rate is considered. The radar measurements are corrupted with zero-mean errors that are Gaussian distributed and have standard deviations of 3 m, 1.1 mrad, and 1.1 mrad in range, azimuthal angle, and vertical angle, respectively. An extended Kalman filter is used for tracking and the filter gains are computed with the standard equations. The process noise power spectral density (PSD) of the tracking system is $q = 100\ \mathrm{m}^2/\mathrm{s}^3$ in each coordinate. This process noise PSD is selected based on the maximum expected acceleration of the target [20], which is 40 m/s^2. Selecting a larger process noise PSD results in a smaller bias during a maneuver and a state error covariance that is large (and wrong) when the target is not maneuvering. Selecting a smaller process noise PSD results in a larger bias during a maneuver and a state error covariance that is too small when the target is maneuvering.

Given the trajectory (i.e., position, velocity, and acceleration) for sensor and target and prediction codes configured with the same characterization of the tracking system described above, the SNO covariance and bias can be computed. The sensor errors in the cross range coordinate are dependent on range as reflected in the computation of the tracking index. For range, the tracking index is

$$\Gamma^r = \sqrt{\frac{\sigma_v^2 T^3}{\sigma_r^2}} \tag{45}$$

where $\sigma_r = 3$ m. For cross range, the tracking index is given by

$$\Gamma_k^{cr} = \sqrt{\frac{\sigma_v^2 T^3}{r_k^2 \sigma_\theta^2}} \tag{46}$$

where r_k is the range from sensor to target at time t_k and $\sigma_\theta = 1.1$ mrad. Using these tracking indices, measurement variances, track rate, and number of measurements, the gains and error covariance can be approximated for each coordinate in the sensor frame. In other words, the SNO covariance and bias as well as the track filter covariance are generated in the orthogonal coordinates that are aligned with the sensor frame. Given r_k, the target acceleration at t_k, the track rate, the number of measurements used in the track, and independence of the errors in each of the sensor coordinates, the predicted RMSE as a function of time can be computed for each of the sensor coordinates using the results from (35) and (36) for position and velocity, respectively. The predicted RMSE that includes all coordinates is computed by adding the mean squared error of all coordinates before taking the square root.

The single sensor, single target scenario used in this analytic computation of RMSE is shown in Fig. 1, where only Sensor 1 is active. In this scenario, a target maneuvers with a near constant acceleration turn of 20 m/s^2 between the intervals $242 \le t_k \le 261$ s (Maneuver 1) and $461 \le t_k \le 479$ s (Maneuver 2), and a near constant acceleration turn of 40 m/s^2 between the interval $961 \le t_k \le 977$ s (Maneuver 3). The predicted performance results for Sensor 1 are shown in Figs. 2 and 3 and compared with the average results from Monte Carlo simulation with 50 runs. Note that predicted results match rather closely to those averages from the

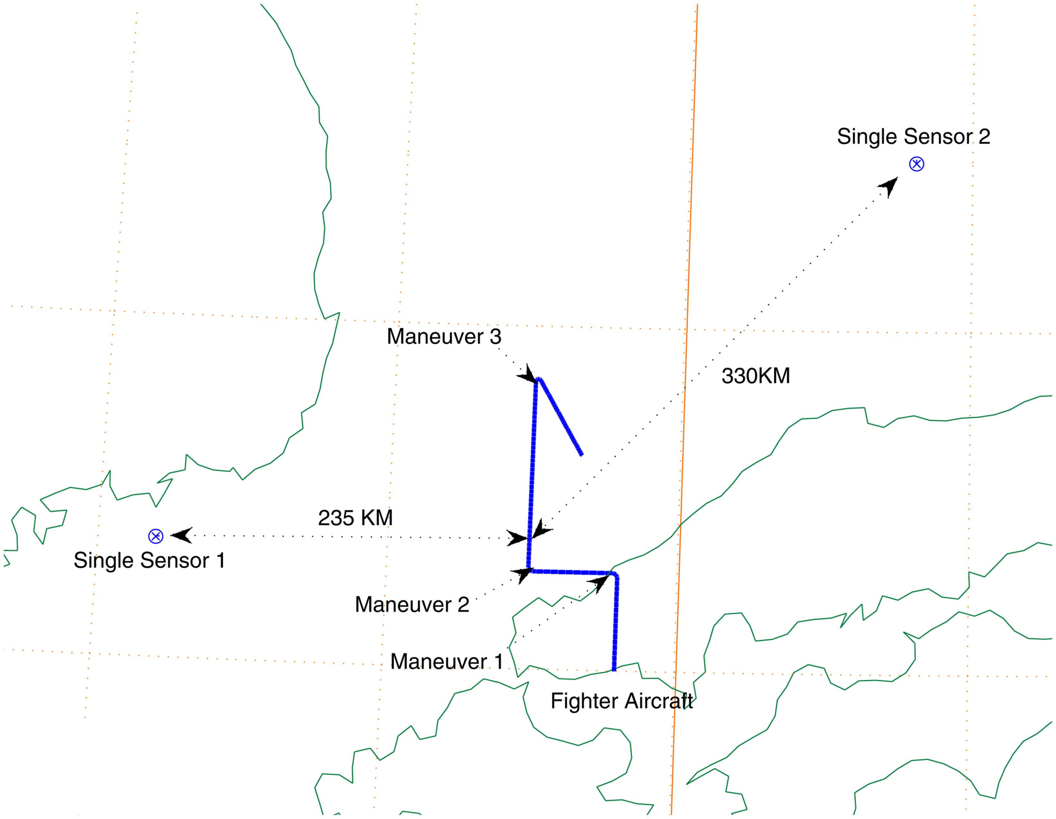


Fig. 1. Multiple sensor and single target scenario.

Monte Carlo simulations. The track filter covariance was also used to predict the RMSE by taking the square root of the sum of the diagonal elements of the covariance matrix for position and velocity.

Comparing Figs. 4–5 highlights an additional improvement for modeling the filter settling time. Since the prediction method is a function of the true target acceleration, the response to maneuvers is assumed instantaneous while in the simulations the response is not instantaneous, and thus one expects the predicted "jump" in RMSE during a maneuver to lead that of the simulated results. Further, one expects the difference between predicted and simulated results to be noticeable when the acceleration in a given coordinate quickly spikes up. Figure 4 illustrates the divergence for maneuver 2 where the target acceleration in cross range quickly spikes and then gradually tapers off. To model the filter settling time, a moving average can be run on the RMSE of the predicted results as shown in Fig. 5. The moving average implemented gives the current predicted RMSE as an average of the current update and the four previous updates. The number of updates chosen for the moving average was selected based on the expected filter settling time given the filter parameters and update rate [20]. Note that all figures illustrating predicted performance, with the exception of Fig. 4, were generated using a moving average as part of the prediction codes.

Figures 2–5 illustrate the inability of the track covariance to accurately predict filter performance. During steady-state tracking with no target maneuver, the track covariance indicates that the position errors are 10% larger than simulation results. In addition, the track covariance indicates that the position errors are significantly smaller than simulation results when the target does maneuver. The same result holds true for velocity as well. The performance prediction of errors in velocity by the track filter covariance has significant error when the target is not maneuvering and this is due to the closer coupling of the random acceleration error modeled to velocity than position. Similarly, the predicted performance versus simulation results for Sensor 2 are shown in Figs. 6 and 7. In this case, the predicted errors also match quite well. However, as illustrated in Fig. 8, we have not modeled the filter setting time as accurately as we did with Sensor 1. In the sensor coordinate frame for Sensor 2, the target acceleration spikes negative before quickly tapering positive. In this case, the negative cross range acceleration did not last long enough for the filter in the simulations to settle. In addition, the quick tapering in the opposite direction had the effect of reducing the maximum lag that was developing in negative cross range coordinate.

4. PREDICTING PERFORMANCE FOR MULTISENSOR TRACKING OF A MANEUVERING TARGET

A method for predicting the performance of a multisensor system is developed in this section. The result

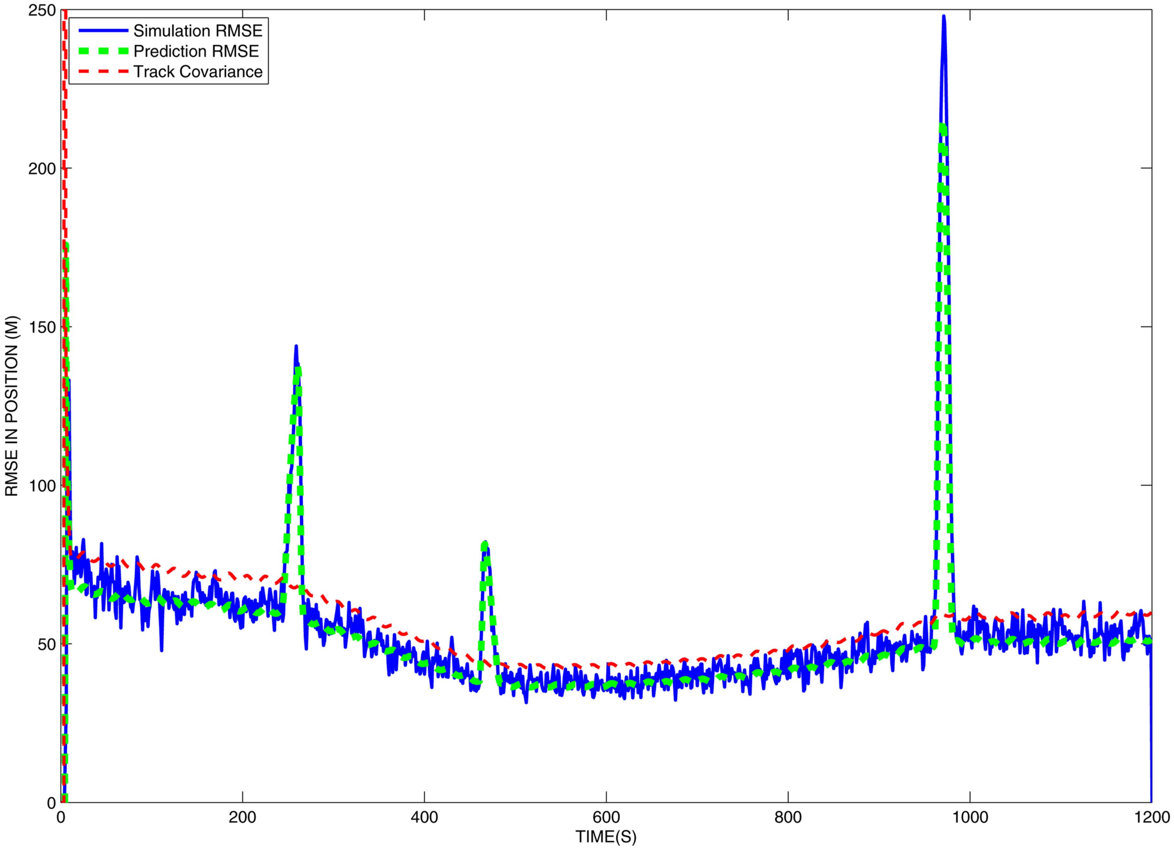


Fig. 2. RMSE in position for Sensor 1.

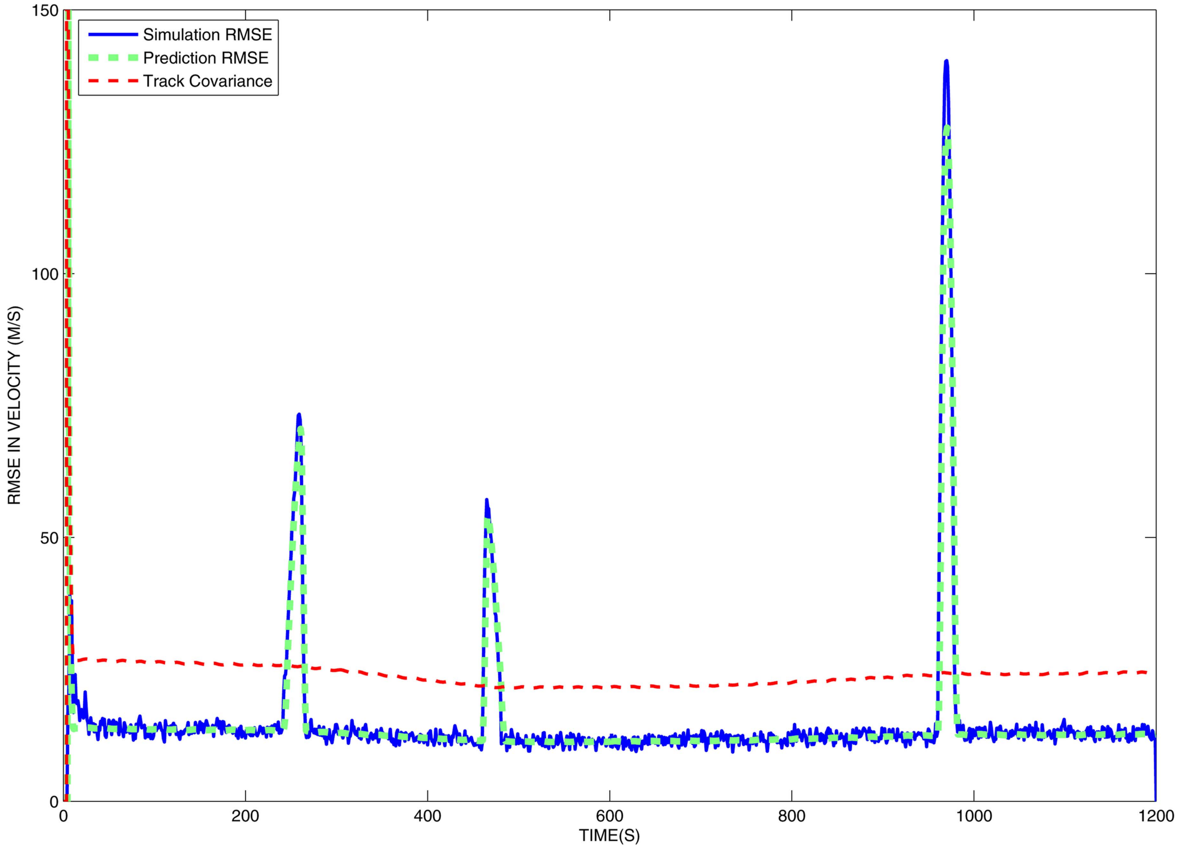


Fig. 3. RMSE in velocity for Sensor 1.

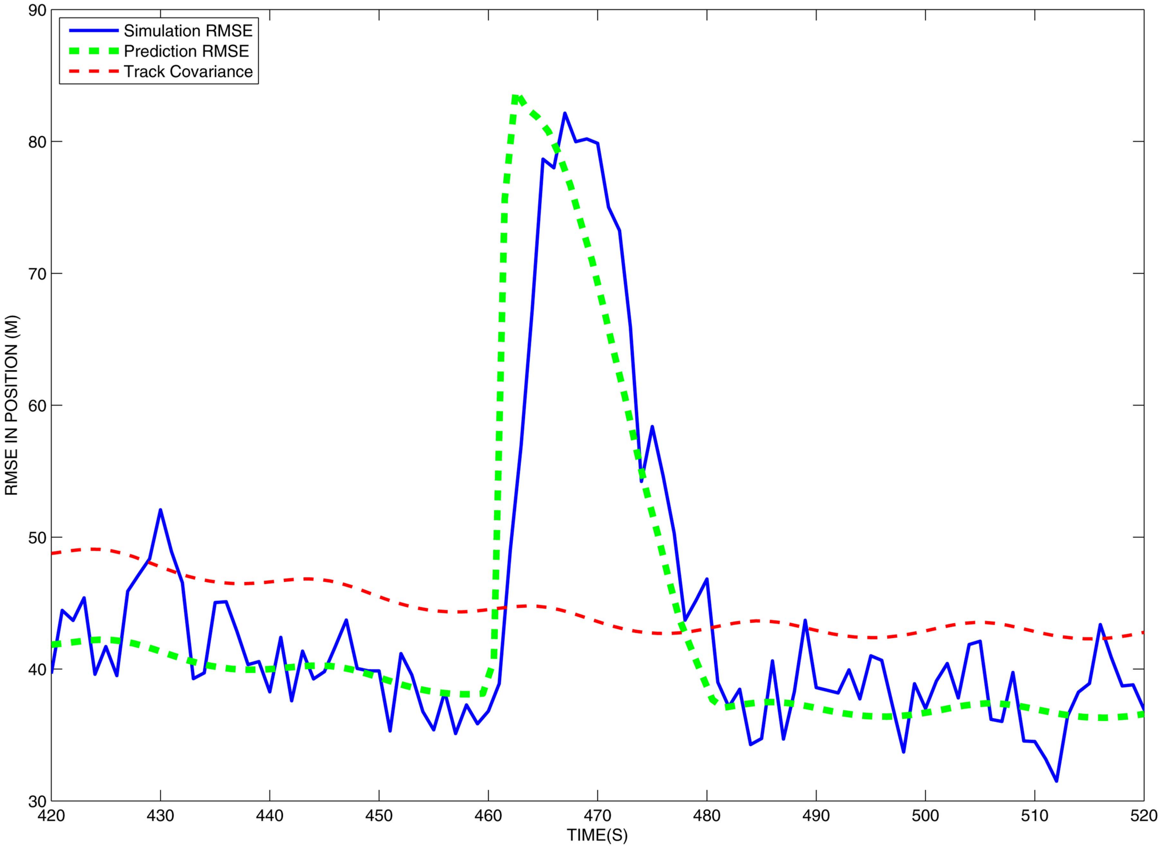


Fig. 4. RMSE during maneuver 2 for Sensor 1 with no moving average applied.

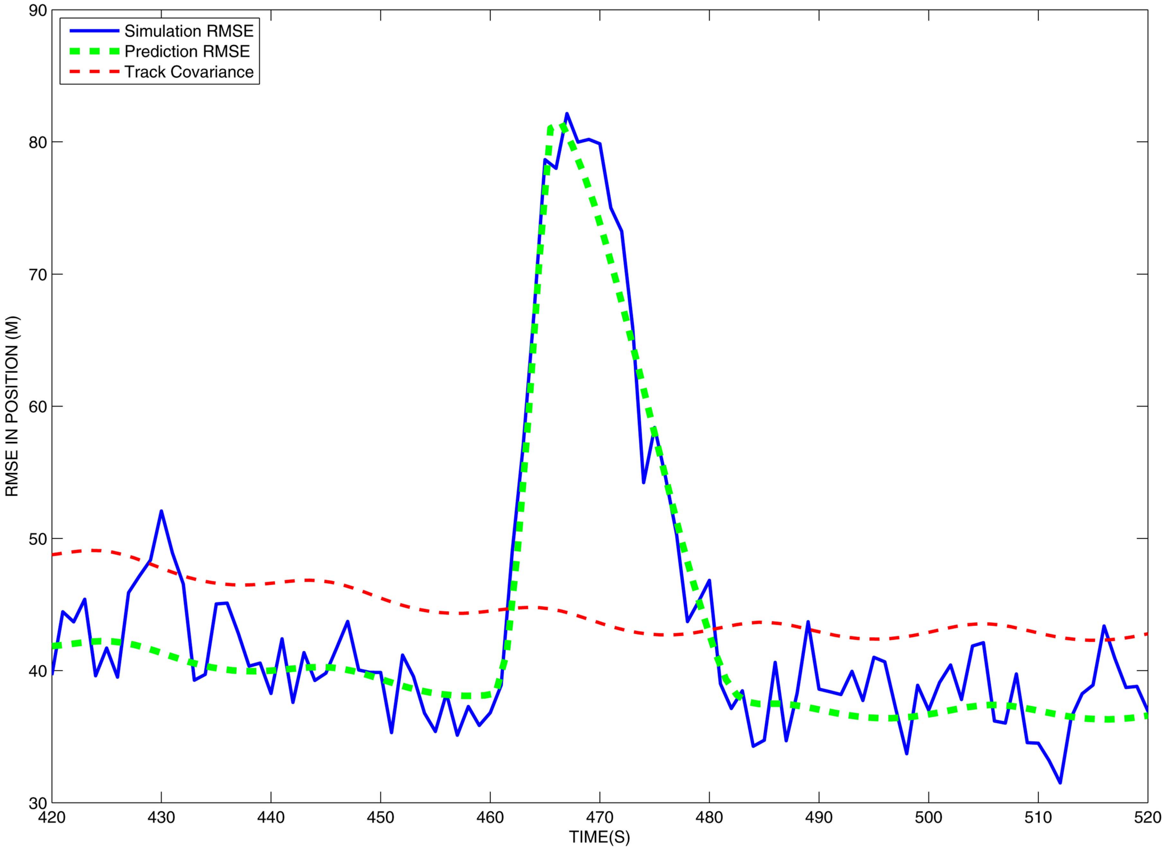


Fig. 5. RMSE during maneuver 2 for Sensor 1 with 5 sample moving average applied.

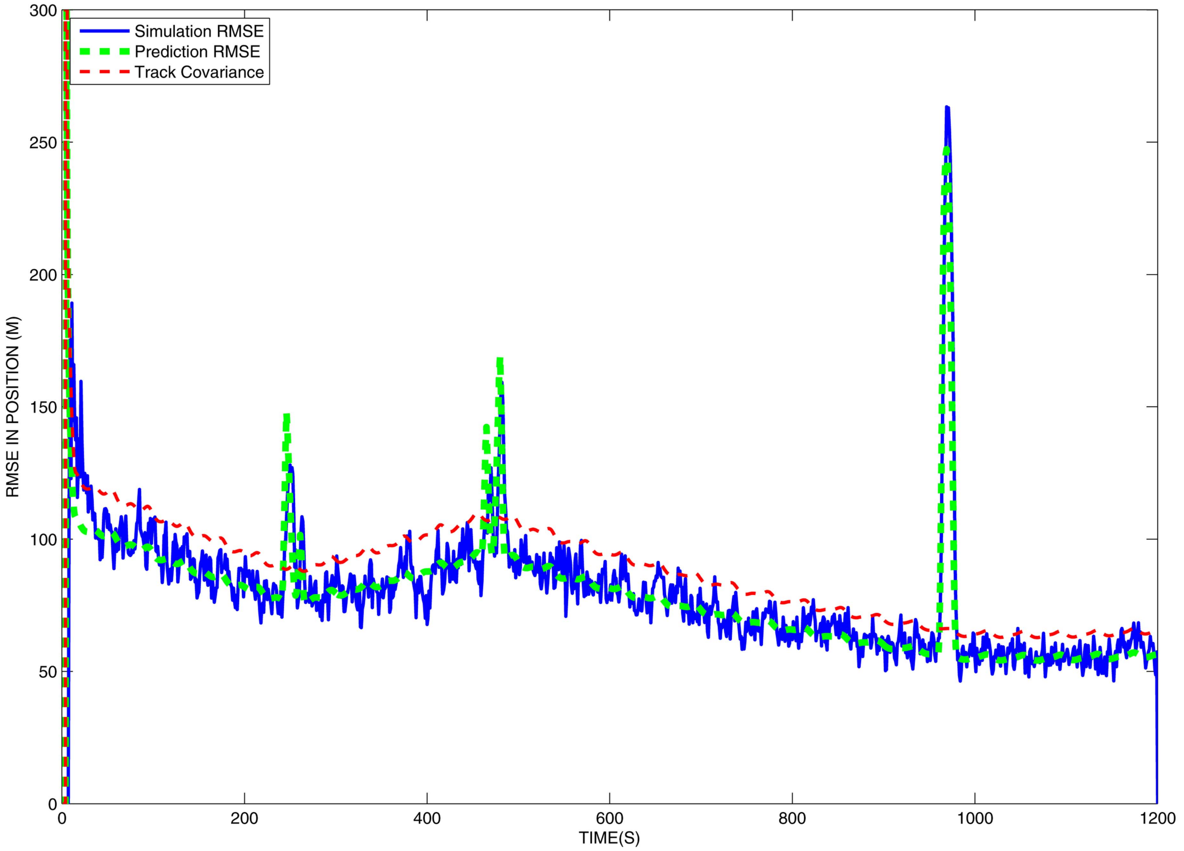


Fig. 6. RMSE in position for Sustain 2.

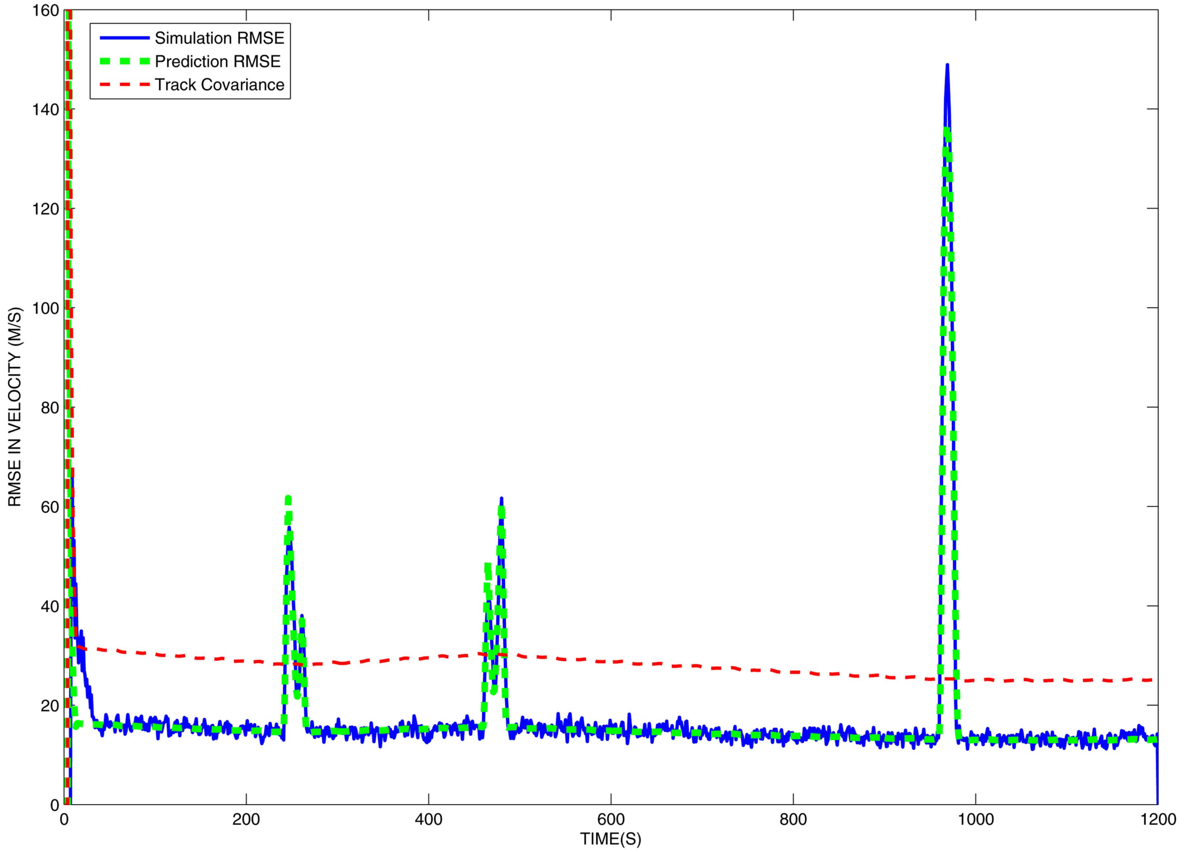


Fig. 7. RMSE in velocity for Sensor 2.

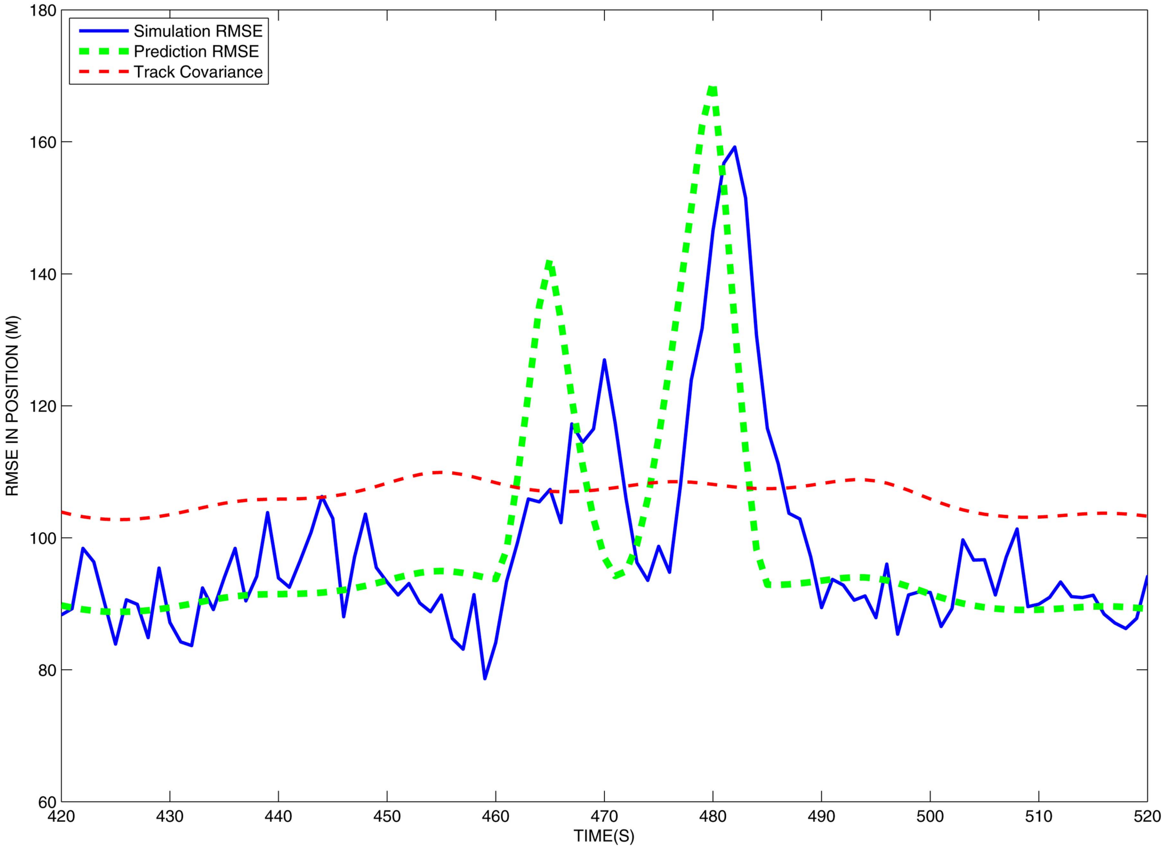


Fig. 8. RMSE during maneuver 2 for Sensor 2.

in (40) gives a direct relationship between the state estimate $X_{k|k}$, and the tracks from sensor i, $X^i_{k|k}$. Now consider a linear system of N independent track reports of the same target from N sensors dispersed on an oblate earth. Then,

$$\begin{bmatrix} X^1_{k|k} \\ X^2_{k|k} \\ \vdots \\ X^N_{k|k} \end{bmatrix} = \begin{bmatrix} M^1_k \\ M^2_k \\ \vdots \\ M^N_k \end{bmatrix} X_k + \begin{bmatrix} M^1_k L^1_k \\ M^2_k L^2_k \\ \vdots \\ M^N_k L^N_k \end{bmatrix} + \begin{bmatrix} W^1_k \\ W^2_k \\ \vdots \\ W^N_k \end{bmatrix} \tag{47}$$

where X_k is the unknown target state at time t_k, M^i_k is the rotation matrix of the state vector to sensor i that is aligned with a common tracking frame, L^i_k is the translation to sensor i from the common tracking frame in that frame, and W^i_k is the error in the track estimate from sensor i with covariance $P^i_{k|k}$. Assuming the errors in the tracks are independent, then

$$Y = \begin{bmatrix} X^1_{k|k} \\ X^2_{k|k} \\ \vdots \\ X^N_{k|k} \end{bmatrix} - \begin{bmatrix} M^1_k L^1_k \\ M^2_k L^2_k \\ \vdots \\ M^N_k L^N_k \end{bmatrix} \tag{48}$$

$$H = [M^1_k \quad M^2_k \cdots M^N_k] \tag{49}$$

$$\begin{aligned} H^T R^{-1} H &= [(M^1_k)^T \quad (M^2_k)^T \cdots (M^N_k)^T] \\ &\quad \times \begin{bmatrix} (P^1_{k|k})^{-1} & 0 & \cdots & 0 \\ 0 & (P^2_{k|k})^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & (P^N_{k|k})^{-1} \end{bmatrix} \begin{bmatrix} M^1_k \\ M^2_k \\ \vdots \\ M^N_k \end{bmatrix} \\ &= \sum_{i=1}^{N} (M^i_k)^T (P^i_{k|k})^{-1} M^i_k. \end{aligned} \tag{50}$$

Substituting this result back into (40) yields the equation for the least squares estimate of a fused track.

$$\begin{aligned} X_{k|k} &= \left[\sum_{i=1}^{N} (M^i_k)^T (P^i_{k|k})^{-1} M^i_k \right]^{-1} \\ &\quad \times \left[\sum_{i=1}^{N} (M^i_k)^T (P^i_{k|k})^{-1} (X^i_{k|k} - M^i_k L^i_k) \right]. \end{aligned} \tag{51}$$

Given a set of measurements with a variable number of Cartesian dimensions, a "coordinate pickoff" matrix of ones and zeros can be applied to each M^i_k so that the measurements for each sensor i are only a function of the state in the cartesian coordinates of interest. This method may be applied to a multiple sensor system that includes sensors with an arbitrary combination of one, two, or three dimensional cartesian measurements.

Array face coordinates is a cartesian system with origin at the sensor array face. In the native sensor frame, some radars measure range (r) and two angles commonly known as u and v. Since the errors in each sensor coordinate are assumed to be independent, the track covariance in orthogonal coordinates of the sensor frame (e.g., array face coordinates) can be approximated in a block diagonal form. The filter error covariance of each coordinate can be approximated with the alpha-beta filter covariance (19) with σ_w^2 representing the variance of the measurement errors in either the range or cross range coordinate. Thus, given the sensor location and measurement rate, measurement error variance for each coordinate, and process noise variance, the track indices can be defined for the range coordinate and the cross range coordinates. The error covariance for a track estimate from a three dimensional sensor i is represented as

$$P_{k|k}^i = \begin{bmatrix} P_{k|k}^r & 0_{2\times 2} & 0_{2\times 2} \\ 0_{2\times 2} & P_{k|k}^u & 0_{2\times 2} \\ 0_{2\times 2} & 0_{2\times 2} & P_{k|k}^v \end{bmatrix} \qquad (52)$$

where it is assumed that the native measurement frame of the sensor is range (r) and two angles (u and v) and the covariance matrix for each coordinate is derived from (19) with σ_w^2 denoting the variance of the errors in either the range or cross range coordinates. The covariance is defined in a Cartesian coordinate system placed at the sensor with the axes aligned with the u and v measurements and the range vector to the target. The rotational transform and translation to sensor i, M_k^i and L_k^i, includes the sensor-to-local transformation as well as the transformation from sensor i to the common tracking frame. Embedded in this transformation is a rotation of the covariance in (52) into array face coordinates. For a track that requires prediction for time alignment before fusion, the standard covariance prediction equations of the Kalman filter can be used to compute the covariance for an m-step ahead prediction.

4.1. Bias of a Fused Track for a Deterministic Maneuver

Consider

$$\begin{aligned} X_{k|k} - X_k &= P_{k|k}^f \left[\sum_{i=1}^{N} (M_k^i)^T (P_{k|k}^i)^{-1} (X_{k|k}^i - M_k^i L_k^i) \right] - X_k \\ &= P_{k|k}^f \left[\sum_{i=1}^{N} (M_k^i)^T (P_{k|k}^i)^{-1} (X_{k|k}^i - M_k^i L_k^i - M_k^i X_k) \right] \\ &= P_{k|k}^f \left[\sum_{i=1}^{N} (M_k^i)^T (P_{k|k}^i)^{-1} (X_{k|k}^i - X_k^i) \right] \qquad (53) \end{aligned}$$

where

$$P_{k|k}^f = \left[\sum_{i=1}^{N} (M_k^i)^T (P_{k|k}^i)^{-1} M_k^i \right]^{-1} \qquad (54)$$

and X_k^i is the true target state in the coordinate frame of sensor i. Taking the expected value and substituting an estimate of the bias for the fused track as

$$B_{k|k}^f = E[X_{k|k} - X_k] = P_{k|k}^f \left[\sum_{i=1}^{N} (M_k^i)^T (P_{k|k}^i)^{-1} B_{k|k}^i \right] \qquad (55)$$

where $B_{k|k}^i$ is the bias or maneuver lag in the track for sensor i that results from a maneuvering target. Thus, the biases and corresponding maneuver lags fuse with a gain proportional to the inverse of the covariance for the individual sensors. The bias of each coordinate of the track can be approximated with the bias or maneuver lag of the alpha-beta filter given by (30) with A_k representing the acceleration of the target in either the range or cross range coordinate for each sensor. The α and β are those that result from the tracking index. Thus, given the sensor location and measurement rate, measurement error variance for each coordinate, and process noise variance used for tracking in that sensor, the tracking indices can be defined for the range coordinate and the cross range coordinates. Thus, the bias or maneuver lag for a track estimate from a three dimensional sensor i is represented as

$$B_{k|k}^i = \begin{bmatrix} B_{k|k}^r \\ B_{k|k}^u \\ B_{k|k}^v \end{bmatrix} \qquad (56)$$

where it is assumed the native measurement frame of the sensor is range (r) and two angles (u and v) and the bias vector for each coordinate is derived from (30) with A_k denoting the acceleration of the target in the range or cross range coordinates. The bias vector is defined in a Cartesian coordinate system placed at the sensor with the axes aligned with the u and v measurements and the range vector to the target. For a track that requires prediction for time alignment before fusion, (34) can be used to compute the bias or maneuver lag for an m-step ahead prediction and a maneuvering target.

4.2. SNO Covariance for a Fused Track

Taking the expected value of (51) for $\bar{X}_{k|k} = E[X_{k|k}]$ gives

$$X_{k|k} - \bar{X}_{k|k} = P_{k|k}^f \left[\sum_{i=1}^{N} (M_k^i)^T (P_{k|k}^i)^{-1} (X_{k|k}^i - \bar{X}_{k|k}^i) \right]. \qquad (57)$$

Let the SNO covariance of the fused track be denoted as

$$S_{k|k}^f = E[(X_{k|k} - \bar{X}_{k|k})(X_{k|k} - \bar{X}_{k|k})^T]. \qquad (58)$$

Assuming sensor errors in the sensor tracks are independent and taking the expected value, the SNO covariance of a fused track is approximately by

$$S_{k|k}^{f} = P_{k|k}^{f}\left[\sum_{i=1}^{N}(M_k^i)^T(P_{k|k}^i)^{-1}S_{k|k}^i(P_{k|k}^i)^{-1}M_k^i\right](P_{k|k}^{f})^T \tag{59}$$

where $S_{k|k}^i$ is the SNO covariance for sensor i. Since the errors in each sensor coordinate tend to be independent, the SNO covariance in sensor coordinates can be approximated in a block diagonal form. The SNO covariance of each coordinate can be approximated with the SNO covariance of the alpha-beta filter (29) with σ_w^2 representing the variance of the measurement errors in either the range or cross range coordinate for each sensor. The α and β are those that results from the tracking index. Thus, given the sensor location and measurement rate, measurement error variance for each coordinate, and process noise variance used for tracking in that sensor, the tracking index can be defined for the range coordinate and the cross range coordinates. The SNO covariance for a track estimate from a three dimensional sensor i is represented as

$$S_{k|k}^i = \begin{bmatrix} S_{k|k}^r & 0_{2\times 2} & 0_{2\times 2} \\ 0_{2\times 2} & S_{k|k}^u & 0_{2\times 2} \\ 0_{2\times 2} & 0_{2\times 2} & S_{k|k}^v \end{bmatrix} \tag{60}$$

where it is assumed the native measurement frame of the sensor is range (r) and two angles (u and v) and the sensor-noise only covariance matrix for each coordinate is derived from (29) with σ_w^2 denoting the variance of the errors in either the range or cross range coordinates. The covariance is defined in a Cartesian coordinate system placed at the sensor with the axes aligned with the u and v measurements and the range vector to the target. For a track that requires prediction for time alignment before fusion, (33) can be used to compute the SNO covariance for an m-step ahead prediction.

5. EXAMPLE OF MULTISENSOR TRACKING FOR A MANEUVERING TARGET

As a numerical example of predicted performance for multiple sensors, consider two independent systems with a 1 Hz measurement rate. The radar measurements are corrupted with zero-mean errors that are Gaussian distributed and have standard deviations of 3 m, 1.1 mrad, and 1.1 mrad in range, azimuthal angle, and vertical angle, respectively. The location of each sensor is shown in Fig. 1. The multisensor tracking system is measurement based with measurements from two unbiased sensors shared over an ideal communications link. Measurement-to-track association performs perfectly given this particular multisensor and single target scenario. The composite tracker employs an extended Kalman filter with nearly constant velocity and process noise PSD of $q = 100\ \mathrm{m}^2/\mathrm{s}^3$ in each coordinate. At each scoring time, the RMSE of the filter state is computed and averaged over all Monte Carlo runs.

The tracking system configuration in the prediction codes is identical to that of the simulations. At each time step in the prediction codes, we compute the steady state error covariance for each sensor as shown in (54). The magnitude of the diagonals for the actual filter track covariance is also displayed in the plots to illustrate the inability of the track filter covariance to accurately predict track filter performance.

The predicted performance results for position and velocity are compared in Figs. 9 and 10 respectively to the results from Monte Carlo simulations with 25 runs. From these figures, it is evident the performance prediction methods presented in this work are more accurate than using the track filter covariance to predict performance. These figures also illustrate the improvement in tracking due to the geometric diversity of the sensors. As predicted, the simulations show the bias due to the maneuver lag is significantly reduced during maneuvers one and two. Due to the short time of these two maneuvers and the geometric diversity of the sensors, the bias in position for maneuvers one and two is insignificant.

Figures 9 and 10 also illustrate the inability of the track filter covariance to predict performance. Consistent with the single sensor case, the covariances in both position and velocity are larger than simulation results when the target is not maneuvering and smaller than simulation results when the target is maneuvering. Finally, although it is noted the predicted performance is slightly optimistic when compared to the simulated results during periods the target is not maneuvering, the predicted performance is still superior to the performance as predicted by the track filter covariance.

Figures 11 and 12 compare the predicted performance results from the single radar tracking example in Section 3 to the predicted performance of two independent radars fusing measurements for a single track. During non maneuvering times, it is apparent from the figures that a variance reduction benefit is obtained from the measurement fusion. The bottom plot in Fig. 12 highlights the improvement in variance reduction for the velocity.

Given a set of circumstances whereby it is known generally where a threat will be launched and what area protection is required, the performance prediction methods presented in this work can be used with predicted trajectories and a desired level of performance to determine optimal placement of sensor resources. Further, this method easily extends to any number of sensors and therefore can be used to help determine the number of sensors required to meet a desired level of performance.

6. CONCLUSIONS

Given the sensor location, target location and accelerations as a function of time, the sensor parameters,

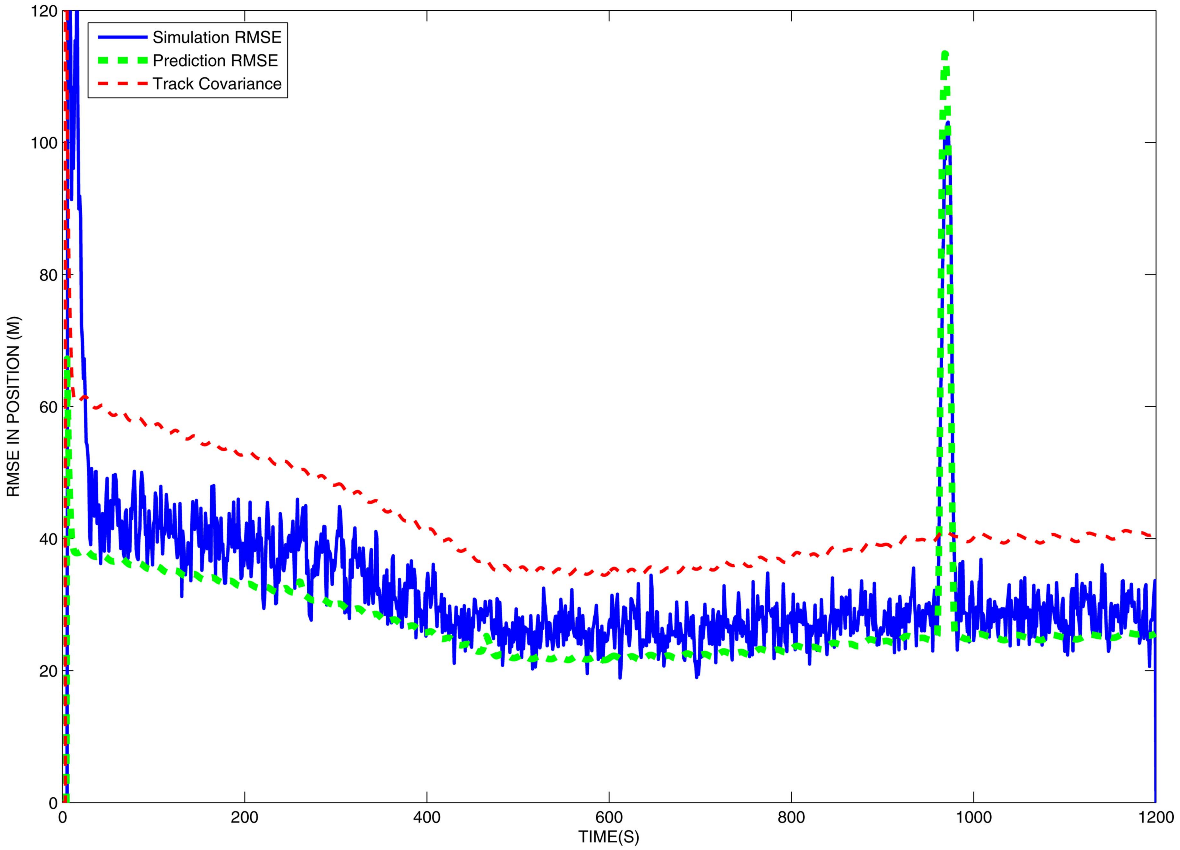


Fig. 9. RMSE in position for fused track from two sensors.

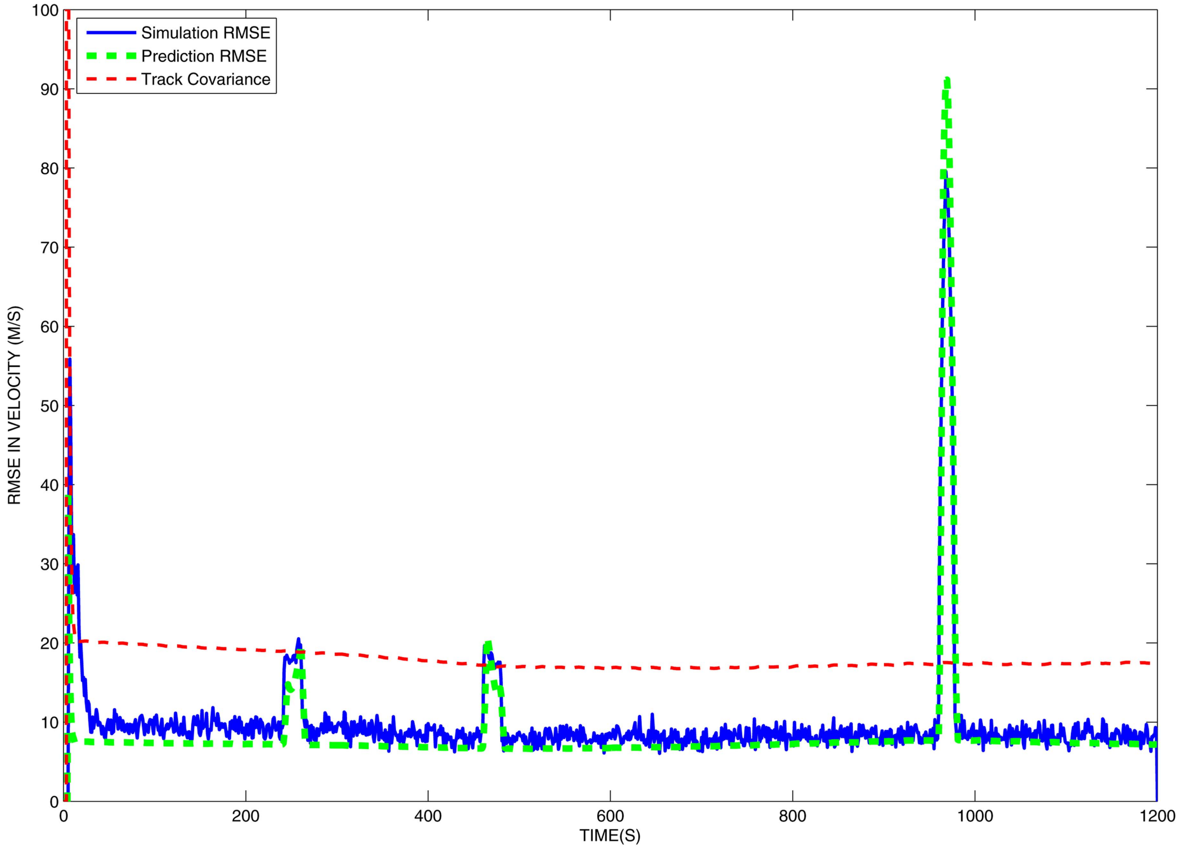


Fig. 10. RMSE in velocity for fused track from two sensors.

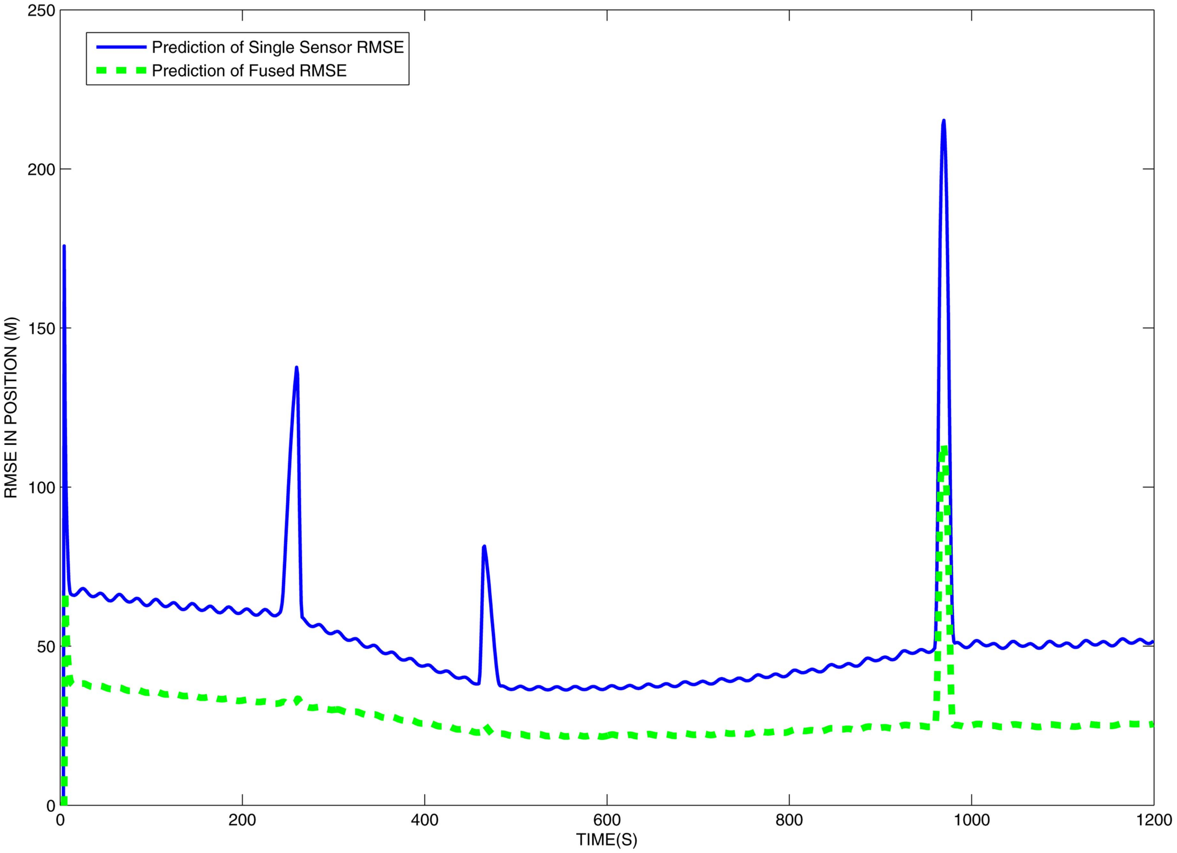


Fig. 11. Predicted RMSE in position for a track from single radar versus a fused track from two radars.

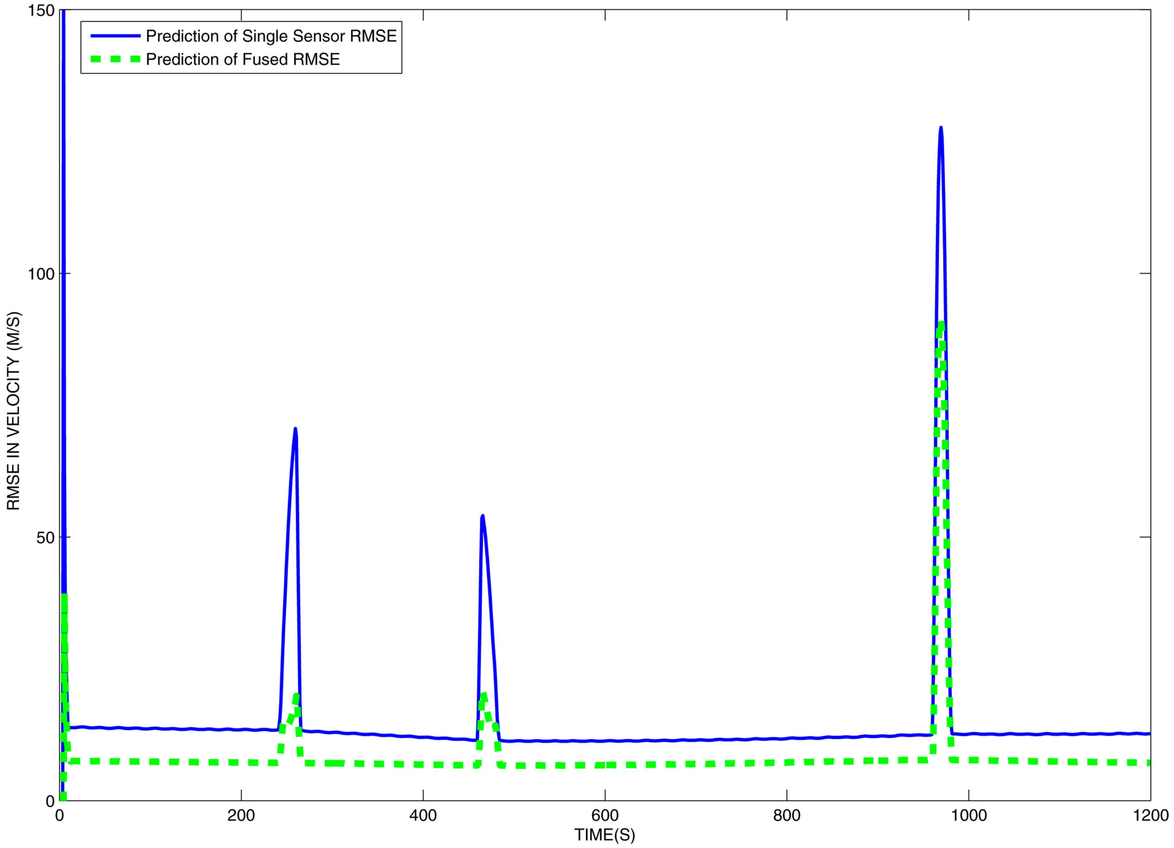


Fig. 12. Predicted RMSE in velocity for a track from single radar versus a fused track from two radars.

and the process noise covariance, it has been shown that one can predict track filtering performance of a multisensor system versus a maneuvering target. Since data association errors are ignored, the predicted performance should be considered as a lower bound for nearly constant velocity tracking. However, the predicted performance does not reflect the improvements in the track filtering that is expected from the use of the IMM estimator [1]. The predicted performance reflects that expected from nearly constant velocity track filtering. Variability between predicted and simulated results can be attributed to more complex modeling in the Monte Carlo simulation environment. While synchronized detections have been assumed in performance prediction methods, detections in simulations for multiple radars do not have synchronized dwell times due to missed detections. In addition, simulations have randomized starting times and logic devoted to resource allocation. Furthermore, complex simulations model performance degrading effects related to off beam centered tracking when targets enter regions near the radar field of view limits. While the predicted results do an remarkable job of capturing the effects of off-broadside tracking, the same can not be said for modeling the effects of off-beam center tracking. However, such conditions under which predicted results diverge from simulated results are understood and therefore, it is easy to detect and display them with the predicted data. The availability of this additional information allows for more accurate interpretation of the predicted performance data. Therefore, the predicted results offer a very reasonable confirmation of the simulated performance results and provide an analytical basis for use in real-time command and control processes and system engineering.

REFERENCES

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

[2] Y. Bar-Shalom and W. D. Blair (Eds.)
Multitarget-Multisensor Tracking: Applications and Advances III.
Norwood, Massachusetts: Artech House, 2000.

[3] T. Ender, R. Leurck, B. Weaver, P. A. Miceli, W. D. Blair, P. D. West, and D. Mavris
Systems-of-systems analysis of ballistic missile defense architecture effectiveness though surrogate modeling and simulation.
In *Proceedings of the 2nd IEEE International Systems Conference*, **23** (Apr. 2008), 333–340.

[4] P. Tichavsky, C. H. Muravchik, and A. Nehorai
Posterior Cramer-Rao bounds for discrete-time nonlinear filtering.
IEEE Transactions on Signal Processing, **46** (May 1998), 1386–1396.

[5] A. Farina, S. Immediata, B. Ristic, and L. Timmoneri
CRLB with $P_d < 1$ for fused tracks.
2005 8th International Conference on Information Fusion, July 2005, 25–28.

[6] C. Hue, J-P. Le Cadre, and P. Perez
Posterior Cramer-Rao bounds for multi-target tracking.
IEEE Transactions on Aerospace and Electronic Systems, **42**, 1 (Jan. 2006), 37–49.

[7] M. L. Hernandez, B. Ristic, and A. Farina
A performance bound for manoeuvring target tracking using best-fitting Gaussian distributions.
2005 7th International Conference on Information Fusion, July 2005, 1–8.

[8] X. R. Li and Y. Bar-Shalom
A hybrid conditional averaging technique for performance prediction of algorithms with continuous and discrete uncertainties.
Proceedings 1994 American Control Conference, June 1994, 1530–1534.

[9] X. R. Li and Y. Bar-Shalom
Performance prediction of the interacting multiple model algorithm.
IEEE Transactions on Aerospace and Electronic Systems, **29**, 3 (July 1993), 755–770.

[10] C. W. Frei and L. Y. Pao
Non-simulation performance prediction methods for different implementations of a multisensor fusion algorithm.
Proceedings 13th World Congress of the International Federation of Automatic Control, July 1996, 269–274.

[11] X. R. Li and Y. Bar-Shalom
Stability evaluation and track life of the PDAF for tracking in clutter.
IEEE Transactions on Automatic Control, **36**, 5 (May 1991), 588–602.

[12] R. W. Osborne and W. D. Blair
Update to the hybrid conditional averaging performance prediction of the IMM algorithm.
IEEE Transactions on Aerospace and Electronic Systems, **47**, 4 (Oct. 2011), 2967–2974.

[13] W. D. Blair
Fixed-gain, two-stage estimators for tracking maneuvering targets.
Technical report NSWCDD/TR-92/297, Naval Surface Warfare Center Dahlgren Division, Dahlgren, VA, July 1992.

[14] W. D. Blair
Fixed-gain, two-stage estimators for tracking maneuvering targets.
IEEE Transactions on Aerospace and Electronic Systems, (July 1993), 1004–1014.

[15] W. D. Blair and Y. Bar-Shalom
Tracking maneuvering targets with multiple sensors: Does more data always mean better estimates?
IEEE Transactions on Aerospace and Electronic Systems, **32** (Jan. 1996), 450–456.

[16] R. Chen and W. D. Blair
Aeolotropic filter design and measurement gating for remote tracking with gridlock error.
Proceedings of 29th IEEE Southeastern Symposium on System Theory, Cookeville, TN, Mar. 1997.

[17] O. E. Drummond, W. D. Blair, G. C. Brown, T. L. Ogle, Y. Bar-Shalom, R. L. Cooperman, and W. H. Barker
Performance assessment and comparison of various tracklet methods for maneuvering targets.
Signal Processing, Sensor Fusion, and Target Recognition XII, **5096** (2003), 1–26.

[18] V. Jain and W. D. Blair
Tracking with LFM waveforms.
IEEE Transactions on Aerospace and Electronic Systems, (Apr. 2009).

[19] J. E. Gray and W. Murray
The response of the transfer function of an alpha-beta filter to various measurement models.
Proceedings Southeastern Symposium on System Theory, **23** (Mar. 1991), 389–393.

[20] W. D. Blair
Design of nearly constant velocity track filters for tracking maneuvering targets.
11th International Conference on Information Fusion, July 2008, 1–7.

[21] A. Gelb
Applied Optimal Estimation.
The M.I.T. Press, Cambridge, MA, 1974.

W. Dale Blair received the B.S. and M.S. degrees in electrical engineering from Tennessee Technological University, Cookeville, in 1985 and 1987, and the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, in 1998.

He is a principal research engineer with the Georgia Tech Research Institute (GTRI) and currently serves as the technical director for the C2BMC Knowledge Center of the Missile Defense Agency (MDA). While a graduate research assistant, Dr. Blair performed robotic controls research for the Center of Excellence of Manufacturing Research and Technology Utilization at Tennessee Technological University. In 1987, he joined the Naval Systems Division of FMC Corporation in Dahlgren, VA as an electrical engineer, where his work involved the development and evaluation of new algorithms for weapons control. In 1990, Dr. Blair joined the Naval Surface Warfare Center, Dahlgren Division (NSWCDD) in Dahlgren, VA. At NSWCDD, he originated two benchmark problems for target tracking and radar resource allocation, which served as themes for invited sessions at the 1994 and 1995 American Control Conferences. He also led a project that demonstrated through a real-time tracking experiment that modern tracking algorithms can be utilized to reduce the radar time and energy required by a phased array radar to support surveillance tracking. Since joining GTRI in 1997, he has led a multiorganizational team in the development of multiplatform-multisensor-multitarget benchmarks to both air defense and ballistic missile defense. His projects at GTRI focus mostly the modeling and simulation and algorithm assessment associated with the sensor netting for the battle management, command, and control for the ballistic missile defense system. He also leads projects related multiple input, multiple output (MIMO) radar, advanced radar data processing, and sensor netting for detection of chemical and biological weapons.

Dr. Blair is a Fellow of the IEEE and recipient of the 2001 IEEE Nathanson Award for Outstanding Young Radar Engineer. He served as the Editor for Radar System for *IEEE Transactions on Aerospace and Electronic Systems* (T-AES) 1996–99 and Editor-In-Chief (EIC) for IEEE T-AES from 1999–2005. He is the founding Editor-In-Chief for the *International Journal on Advances in Information Fusion* under the International Society for Information Fusion. Dr. Blair also serves on the Board of Governors of IEEE Aerospace and Electronic Systems Society (AESS) from 1998–2003 and 2005–2010. While a member of the IEEE AESS BoG, he served as Associate Vice President for Publications for 2007–10 and Chair of the Strategic Planning Committee for 2009–10, originated the Target Tracking System Panel, and initiated the Systems Panel for Cyber Security Systems. He has received technical awards that include the NSWCDD Technical Excellence Award in 1991 for his contributions to the development of innovative techniques for target trajectory estimation and prediction and the NSWCDD Independent Exploratory Development Excellence Award in 1993. His research interests include radar signal processing and control, resource allocation for multifunction radars, multisensor resource allocation, tracking maneuvering targets, and multisensor integration and data fusion. Dr. Blair's research is reported in over two hundred articles which include 38 refereed journal articles. He served as a lecturer in 1993, 1994, and 1998 for the UCLA Extension Program in three short courses related to advanced topics in target tracking. He recently originated and coordinates three short courses, *Target Tracking in Sensor Systems*, *Target Tracking Concepts*, and *Advanced Target Tracking for Ballistic Missile Defense*, for the Profession Education Department of the Georgia Institute of Technology. Dr. Blair is coeditor and coauthor of the book, *Multitarget-Multisensor Tracking: Advances and Applications III*, and the author of chapter 19 "Radar Tracking Algorithms" of the new edition of *Principles of Modern Radar*. He is also the host and coordinator of the ONR/GTRI Workshop on Target Tracking and Sensor Fusion for 1998 through 2011.

Paul Miceli received the B.S. degree in electrical engineering from Manhattan College and the M.S. degree in electrical engineering from the Georgia Institute of Technology.

He is a senior research engineer for the Electronics Systems Branch of the Georgia Tech Research Institute. During his time in graduate school, his research focused on the development of signal processing algorithms for the suppression of urban noise in radar warning receivers. Since joining the Georgia Tech research faculty full time in May 2002, he has contributed to numerous programs in applications involving target tracking, radar, signal processing, and modeling and simulation. In addition to infrastructure and algorithm development, Mr. Miceli has also long been involved with applied research studies in the fields of multi-sensor fusion and radar performance prediction.

Profile-Free Launch Point Estimation for Ballistic Targets using Passive Sensors

RATNSINGHAM THARMARASA
THIAGALINGAM KIRUBARAJAN
NANDAKUMARAN NADARAJAH
YAAKOV BAR-SHALOM
THAYANANTHAN THAYAPARAN

This paper considers the estimation of the Launch Points (LP) of ballistic targets from two or more passive satellite-borne sensors by fusing their angle-only measurements. The targets are assumed to have a two-stage boost phase with a free-flight phase between the two stages. Due to the passive nature of the sensors, there is no measurement during the free-flight motion. It is also assumed that measurements are available only after a few seconds from the launch time due to cloud cover. In the literature, profile-based methods have been proposed to estimate the target's launch point and trajectory. Profile-based methods normally result in large errors when there is a mismatch between actual and assumed profiles, which is the case in most scenarios. In this paper, a profile-free method is proposed to estimate the target states at the End-of-Burnout (EOB) and LP. Estimates at the EOB are obtained by using forward-filtering with adaptive model selection based on boost phase changes. The LP estimates are obtained using smoothing followed by backward prediction. Uncertainties in the motion model and the launch time must be incorporated in the backward prediction. The LP estimate and the corresponding error covariance are obtained by incorporating the above uncertainties. Simulation results illustrating the performance of the proposed approach are presented.

Manuscript received July 11, 2011; revised October 14, 2011; released for publication October 17, 2011.

Refereeing of this contribution was handled by Peter Willett.

Authors' addresses: R. Tharmarasa and T. Kirubarajan, McMaster University, Hamilton, ON, Canada, E-mail: (tharman@mail.ece.mcmaster.ca, kiruba@mcmaster.ca); N. Nandakumaran, Curtin University of Technology, Perth WA, Australia, E-mail: (n.nadarajah@curtin.edu.au); Y. Bar-Shalom, University of Connecticut, Storrs, CT, E-mail: (ybs@ee.uconn.edu); T. Thayaparan, Defence Research & Development Canada, Ottawa, ON, Canada, E-mail: (Thayananthan.Thayaparan@drdc-rddc.gc.ca).

1557-6418/12/$17.00 © 2012 JAIF

1. INTRODUCTION

Estimating the Launch Points (LP) of ballistic targets is an important problem in missile defence in order to take action against them. In the literature, profile-based and profile-free methods have been proposed to estimate the states of a target at end-of-burnout (EOB) and/or LP using active or passive sensors [6, 7, 12, 13, 22]. In profile-based methods, most model parameters are assumed to be known [12, 16]. In [12], a profile-based maximum likelihood estimation method was proposed to find the launch point estimate. In [21], estimating the launch parameters using an analytic approximation of the trajectory and an Unscented Kalman Filter (UKF) was analyzed. The profile-based methods will give good results when the model assumptions are accurate. However, when the model is poor, estimates of the profile-based methods will have large errors.

The influence of a priori uncertainties in launch time and trajectory profile on estimation of launch point using spaced-based infrared sensors was analyzed in [5]. In [22], the advantages of profile-free methods for EOB state estimation were explained with examples. The launch and impact point estimation of a ballistic target using radar measurements under different hypotheses on the available prior knowledge was studied in [3]. In that paper, nonlinear batch estimator combined with a recursive multiple model particle filter was proposed. In [10], the advantages of the Particle Filter (PF) over the Kalman Filter (KF) based Interacting Multiple Model (IMM) trackers for the launch point estimation of ballistic targets with single stage boost phase was analyzed. A launch point estimation algorithm using U-D factorization-based Kalman filter and Rauch-Tung-Striebel (RTS) smoother was proposed in [14].

In this paper, a profile-free method is proposed to estimate the LP of a ballistic target that has a two stage boost phase with a free-flight phase between the two phases. Two or more satellite-borne passive sensors, which measure the azimuth and elevation, are available to provide measurements. In most cases, an IMM filter will be a better choice to track a target with maneuvers. However, if the maneuver models and the times at which maneuvers occur are known, then a single model filter with time-varying model can be used instead of IMM. If there is no free-flight phase between two stages, it would be challenging to know the stage changes of the boost phase. Due to the passive nature of the sensors, no measurements will be obtained during the free-flight phase. Hence, the times at which boost phase's stage changes occur are known accurately. In this paper, a profile-free single time-varying model filter is proposed to estimate the states at EOB and LP.

The estimate of the target state at the first measurement time can be improved by performing smoothing with all available measurements. The first measurement will be typically received only after a few seconds from the launch time due to cloud cover. A backward predic-

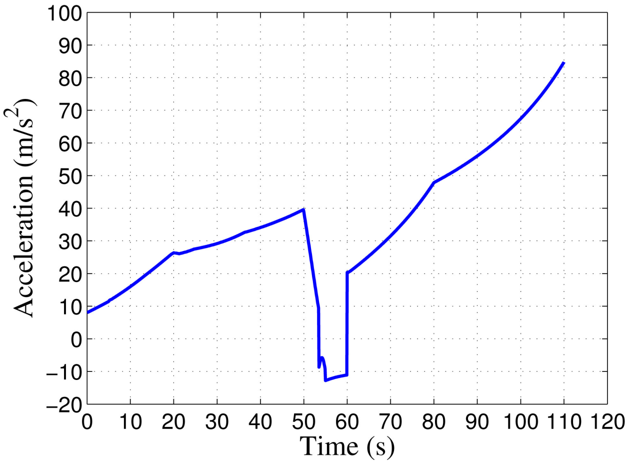


Fig. 1. Acceleration for a sample boost phase.

tion from first measurement time is needed to find the LP estimate. Uncertainties in the launch time and in the dynamic model must be considered while performing backward prediction. A least squares estimation method is proposed for the above backward prediction.

It is always beneficial to estimate the LP as soon as possible. However, the use of measurements over multiple time steps will improve the accuracy of the LP estimate. Due to motion model uncertainties and target maneuvers, measurements beyond a certain time might not add much information to the LP estimate. The Posterior Cramer-Rao Lower Bound (PCRLB) is a useful bound to predict the performance of an estimator [19, 20]. The contribution of the measurements over time to the EOB and LP state estimates can be evaluated using the PCRLB. Based on these PCRLBs, one can decide whether to use all available measurements over time or a subset of them in estimating the LP and EOB states. The PCRLB equations and values for simulations for forward filtering and smoothing are given in this paper.

The remainder of the paper is structured as follows. Section 2 describes the problem considered in this paper. The dynamic models used in the approach are given in Section 3. The profile-free estimation algorithm is explained in Section 4. Section 5 presents simulation results that demonstrate the effectiveness of the approach. Conclusions are presented in Section 6.

2. PROBLEM STATEMENT

The following assumptions are made:

- The target has a two-stage boost phase.
- There is a free-flight phase between the two stages of the boost phase.
- No measurements are available during the free-flight phase.
- No measurements are available for a few seconds from the launch time.

The acceleration magnitudes of a sample boost phase [4] are shown in Fig. 1. In this sample boost phase, the first boost stage is from $t_1 = 0$ to $t_2 = 55$ s and the second boost phase is from $t_3 = 60$ to $t_4 = 110$ s with a free-flight phase from $t_2 = 55$ to $t_3 = 60$ s. There is a continuous reduction in the acceleration from $t_2 - t_d = 50$ to $t_2 = 55$ s, where t_d is the first stage boost decay time.

Measurements are azimuth θ and elevation γ (in radians). The measurement equation for sensor j is given by

$$\mathbf{z}^j(k) = \begin{bmatrix} \theta^j(k) \\ \gamma^j(k) \end{bmatrix} = \underbrace{\begin{bmatrix} \text{atan}\left(\dfrac{y(k) - y_s^j(k)}{x(k) - x_s^j(k)}\right) \\ \text{atan}\left(\dfrac{z(k) - z_s^j(k)}{\sqrt{(x(k) - x_s^j(k))^2 + (y(k) - y_s^j(k))^2}}\right) \end{bmatrix}}_{h(\mathbf{x}_p(k),\mathbf{x}_s^j(k))} + w^j(k) \tag{1}$$

where (x, y, z) and (x_s^j, y_s^j, z_s^j) are the locations of the target and sensor j, respectively, $\mathbf{x}_p = [x\ y\ z]'$ is the target position vector, atan$(\cdot)$ is the four-quadrant arctangent, and $w^j(k)$ is a zero-mean Gaussian random variable with covariance $R^j(k)$.

3. DYNAMIC MODELS

The dynamic equation is given by

$$\mathbf{x}(k+1) = F(k)\mathbf{x}(k) + v(k) \tag{2}$$

where $\mathbf{x}(k)$ is the state vector, $F(k)$ is the state transition matrix, and $v(k)$ is zero-mean white noise with covariance $Q(k)$. The dynamic models used in this paper are given below.

3.1. White Noise Jerk model

This is also called Wiener Process Acceleration (WPA) model. In this model, the state vector comprises position, velocity, and acceleration. The state transition matrix $F(k)$ and process noise covariance $Q(k)$ (in one generic coordinate) are given by (this is the discretized continuous time model, CWPA) [2]

$$F(k) = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

and

$$Q(k) = q_m \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix} \tag{4}$$

where T is the sampling interval[1] at k, i.e., $T = t_{k+1} - t_k$, and q_m is the power spectral density (PSD) of the continuous time process noise (with dimension [length]^2/[time]^5).

[1] The sampling interval is assumed constant, but it can be variable, in which case T_k should be used in (3)–(8).

3.2 Wiener Process Jerk Model

In this model (WPJ), the state vector comprises position, velocity, acceleration, and jerk. The state transition matrix $F(k)$ and process noise covariance $Q(k)$ (in one generic coordinate) are given by (this is the discretized continuous time model, CWPJ)

$$F(k) = \begin{bmatrix} 1 & T & T^2/2 & T^3/6 \\ 0 & 1 & T & T^2/2 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

and

$$Q(k) = q_m \begin{bmatrix} T^7/252 & T^6/72 & T^5/30 & T^4/24 \\ T^6/72 & T^5/20 & T^4/8 & T^3/6 \\ T^5/30 & T^4/8 & T^3/3 & T^2/2 \\ T^4/24 & T^3/6 & T^2/2 & T \end{bmatrix} \tag{6}$$

where q_m is the PSD of the continuous time process noise (with dimension $[\text{length}]^2/[\text{time}]^7$).

3.3. Exponentially Autocorrelated Acceleration Model

This model is also called the Singer model. In this model, the state vector comprises position, velocity, and acceleration. The state transition matrix $F(k)$ and process noise covariance $Q(k)$ (in one generic coordinate) are given by [2, 11]

$$F(k) = \begin{bmatrix} 1 & T & (\alpha T - 1 + e^{-\alpha T})/\alpha^2 \\ 0 & 1 & (1 - e^{-\alpha T})/\alpha \\ 0 & 0 & e^{-\alpha T} \end{bmatrix} \tag{7}$$

and

$$Q(k) = 2\alpha\sigma_m^2 \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix} \tag{8}$$

where the acceleration autocorrelation is $E\{a(t)a(t+\tau)\} = \sigma_m^2 e^{-\alpha|\tau|}$. In the above, σ_m^2 is the instantaneous variance of the acceleration and $1/\alpha$ is the time constant of the target acceleration autocorrelation.

4. PROFILE-FREE ESTIMATION ALGORITHM

4.1. Track Initialization for Asynchronous Sensors

One-point initialization with the first measurement is used to initialize the target. Additional information other than the bearing and elevation measurements is needed to initialize the position in 3-D coordinates with a single measurement. It is reasonable to assume that the target's altitude will only be a few kilometers at the first detection time. Hence, the target altitude, $\hat{z}(1)$, is assumed to be $\hat{z}(1) = h_{\max}/2$ with variance $\sigma_h^2 = h_{\max}^2/12$, where $h_{\max}$ is the possible maximum altitude at the first measurement time. If the first measurement is received from sensor i, then the target's position is initialized as

$$\begin{bmatrix} \hat{x}(1) \\ \hat{y}(1) \\ \hat{z}(1) \end{bmatrix} = \begin{bmatrix} x_s^i(1) + \dfrac{(\hat{z}(1) - z_s^i(1))}{\tan(\gamma^i(1))}\cos(\theta^i(1)) \\ y_s^i(1) + \dfrac{(\hat{z}(1) - z_s^i(1))}{\tan(\gamma^i(1))}\sin(\theta^i(1)) \\ \dfrac{h_{\max}}{2} \end{bmatrix} \tag{9}$$

with covariance

$$P(1) = (\bar{H}^i(1)'\bar{R}^i(1)^{-1}\bar{H}^i(1))^{-1} \tag{10}$$

where $\bar{R}^i(1) = \text{diag}(R^i(1), \sigma_h^2)$ and

$$\bar{H}^i(1) = \frac{\partial \bar{h}(\mathbf{x}_p(1), \mathbf{x}_s^i(1))}{\partial \mathbf{x}_p(1)} \tag{11}$$

is the Jacobian matrix with elements

$$\bar{H}^i(1)(1,1) = -\frac{(\hat{y}(1) - y_s^i(1))}{(d_h^i)^2} \tag{12}$$

$$\bar{H}^i(1)(1,2) = \frac{(\hat{x}(1) - x_s^i(1))}{(d_h^i)^2} \tag{13}$$

$$\bar{H}^i(1)(1,3) = 0 \tag{14}$$

$$\bar{H}^i(1)(2,1) = \frac{-(\hat{z}(1) - z_s^i(1))(\hat{x}(1) - x_s^i(1))}{d_h^i(d_v^i)^2} \tag{15}$$

$$\bar{H}^i(1)(2,2) = \frac{-(\hat{z}(1) - z_s^i(1))(\hat{y}(1) - y_s^i(1))}{d_h^i(d_v^i)^2} \tag{16}$$

$$\bar{H}^i(1)(2,3) = \frac{d_h^i}{(d_v^i)^2} \tag{17}$$

$$\bar{H}^i(1)(3,1) = 0 \tag{18}$$

$$\bar{H}^i(1)(3,2) = 0 \tag{19}$$

$$\bar{H}^i(1)(3,3) = 1 \tag{20}$$

with

$$d_h^i = \sqrt{(\hat{x}(1) - x_s^i(1))^2 + (\hat{y}(1) - y_s^i(1))^2} \tag{21}$$

$$d_v^i = \sqrt{(\hat{x}(1) - x_s^i(1))^2 + (\hat{y}(1) - y_s^i(1))^2 + (\hat{z}(1) - z_s^i(1))^2}. \tag{22}$$

In the above, $\bar{h}(\mathbf{x}_p(1), \mathbf{x}_s^i(1))$ is the nonlinear measurement function for the stacked measurement $[\theta^i(1)\ \gamma^i(1)\ \hat{z}(1)]$

$$\bar{h}(\mathbf{x}_p(1), \mathbf{x}_s^i(1)) = \begin{bmatrix} h(\mathbf{x}_p(1), \mathbf{x}_s^i(1)) \\ \hat{z}(1) \end{bmatrix} \tag{23}$$

where $\hat{z}(1)$ is the measurement (prior information) of target altitude.

The target velocity, acceleration, and jerk estimates are initialized with mean zero and variances $(2v_{\max})^2/12$, $(2a_{\max})^2/12$, and $(2j_{\max})^2/12$, respectively, for the X and Y coordinates, and mean $v_{\max}/2$, $a_{\max}/2$, and $j_{\max}/2$ and variances $v_{\max}^2/12$, $a_{\max}^2/12$, and $j_{\max}^2/12$, respectively, for the Z coordinate, where $v_{\max}$, $a_{\max}$, and $j_{\max}$ are the maximum possible velocity, acceleration and jerk, respectively. In the above, uniform distributions in $[-v_{\max}, v_{\max}]$, $[-a_{\max}, a_{\max}]$, and $[-j_{\max}, j_{\max}]$ are used for the X and Y coordinates, since their initial values can be positive or negative, while $[0, v_{\max}]$, $[0, a_{\max}]$, and $[0, j_{\max}]$ are used for the Z coordinate, since its initial values are always positive.

4.2. Track Initialization for Synchronized Sensors

If the sensors are synchronized, the initial target position is found using the Iterated Least Squares (ILS) estimator by fusing the measurements from all the sensors (configuration III fusion [1]). The estimate at iteration $j+1$ is given by

$$\hat{\mathbf{x}}_p^{j+1}(1) = \hat{\mathbf{x}}_p^j(1) + (\bar{\bar{H}}^{j'}(1)\bar{\bar{R}}^{-1}(1)\bar{\bar{H}}^{j}(1))^{-1}\bar{\bar{H}}^{j'}(1)\bar{\bar{R}}(1)^{-1} \times [\mathbf{z}(1) - \mathbf{h}(\hat{\mathbf{x}}_p^j(1), \mathbf{x}_s(1))] \tag{24}$$

where $\mathbf{z}(\underline{1}) = [\mathbf{z}^1(1)\ \ \mathbf{z}^2(1)\cdots\mathbf{z}^n(1)]'$, n is the number of sensors, $\bar{\bar{R}}(1) = \mathrm{diag}(R^1(1), R^2(1), \ldots, R^n(1))$,

$$\mathbf{h}(\mathbf{x}_p(1), \mathbf{x}_s(1)) = \begin{bmatrix} h(\mathbf{x}_p, \mathbf{x}_s^1) \\ h(\mathbf{x}_p, \mathbf{x}_s^2) \\ \vdots \\ h(\mathbf{x}_p, \mathbf{x}_s^n) \end{bmatrix} \tag{25}$$

and

$$\bar{\bar{H}}^{j}(1) = \left.\frac{\partial \mathbf{h}(\mathbf{x}_p(1), \mathbf{x}_s(1))}{\partial \mathbf{x}_p(1)}\right|_{\mathbf{x}_p(1)=\hat{\mathbf{x}}_p^j(1)} \tag{26}$$

is the Jacobian matrix, which is the stacked matrix with elements of ith block given by (12), (13), (15), and (16).

An initial estimate $\hat{\mathbf{x}}_p^0(1)$ for the ILS estimator is obtained from the intersection of the measurement from any one sensor with the earth's surface, as explained in section *Track Initialization for Asynchronous Sensors*.

The covariance of this (nonlinear) estimator[2] is

$$P(1) = (\bar{\bar{H}}(1)'\bar{\bar{R}}(1)^{-1}\bar{\bar{H}}(1))^{-1} \tag{27}$$

where $\bar{\bar{H}}$ is the last $\bar{\bar{H}}^{j}$ from (26).

The initial target velocity, acceleration, and jerk can also be estimated using a polynomial fit with composite measurements from the first few scans. The recursive filter can be started from $k = 1$ with a larger covariance than the polynomial fit indicates to avoid "double counting" the measurement information.

4.3. Forward Filtering

Since there is no measurement between the two stages of the boost phase, the times at which the boost stages are changing are known. If measurements are obtained during t_1 to t_2 and t_3 to t_4, then it is known that the target is under free-flight phase during t_2 to t_3. Absence of measurements for certain time period helps to know the boost phase transition. Hence, there is no need to use an IMM here.

For asynchronous sensors, an Extended Kalman Filter (EKF) is used to handle the nonlinearity in the measurement equation using sequential updating (configuration IV fusion [1]). For synchronized sensors, the EKF can be replaced with a Kalman Filter by finding composite measurements (i.e., complete target positions) from the azimuth and elevation measurements of all the sensors [1]. Composite measurements can be obtained using the ILS estimator, as explained in section *Track Initialization for Synchronized Sensors*. The target's dynamic model is adaptively changed based on the boost phase changes, which are known from the absence of measurements during the free-flight motion. A Wiener process jerk model is used during $[t_1, t_2 - t_d]$, where t_d is the time the target takes to go to free-flight phase from full acceleration of the first stage of the boost phase (decay time). During $[t_2 - t_d, t_2]$, a white-noise jerk model is used with large process noise, whose variance is calculated based on the estimated acceleration at time $t_2 - t_d$ and the acceleration of free-flight phase ($\ddot{x} = 0$, $\ddot{y} = 0$ and $\ddot{z} = -g$, where g is the gravitational acceleration of the earth, assumed constant). In one generic coordinate, the standard deviation of the process noise is set as $(a_{t_2-t_d} - a_{t_2})/t_d$, where a is the acceleration. A white-noise jerk model with zero (or very small) process noise is used during $[t_2, t_3]$. Again, during $[t_3, t_4]$, a Wiener process jerk model is used.

The exact value of t_d is not known to the tracker. A white noise acceleration model with large process noise will work in $[t_2 - t_d, t_2]$ even if the actual motion is Wiener process jerk model, but it is not true for the converse. Hence, a maximum possible duration is used for t_d.

The estimated target state and its covariance must be modified during the model transitions. For the model transition at $t_2 - t_d$, i.e., Wiener process jerk model to white-noise jerk model, the jerk estimates and the corresponding covariances are removed. For the model transition at t_2, i.e., white-noise jerk model with large process noise to white-noise jerk model with zero or very small process noise, the acceleration estimates are changed to $\ddot{x} = 0$, $\ddot{y} = 0$ and $\ddot{z} = -g$, and the variances of the acceleration estimates are set to zero (or very small). For the model transition at t_3, the acceleration estimates are set to zero and the variances of the acceleration estimates are set to $(a_{\max}/2)^2$. Similarly, the jerk estimates are set to zero with variance $(j_{\max}/2)^2$.

A constrained Kalman Filter can be used to impose a minimum acceleration, which will help improve the tracking accuracy [18]. A directional process noise can

[2] This covariance is based on the CRLB. However, as shown in [17], this estimator is statistically efficient, i.e., the CRLB yields the actual covariance.

also be used based on the estimated velocity to reduce the uncertainty in the target motion.

The Wiener process jerk model can be replaced with an exponentially autocorrelated acceleration model. However, an accurate value of α must be known to get better estimates. The comparison of Wiener process jerk model and autocorrelated acceleration model is given in the simulation section.

The sequential updating technique, i.e., update of the target state with the measurement of one sensor at a time, is used to handle measurements from multiple sensors [1].

4.4. LP Estimation using Smoothing

After getting the estimate at EOB using forward filtering, smoothing is applied to find the estimate of the launch point. The smoothing can be performed back to the first measurement time as follows [2]

$$\hat{\mathbf{x}}(k \mid N) = \hat{\mathbf{x}}(k \mid k) + C(k)[\hat{\mathbf{x}}(k+1 \mid N) - \hat{\mathbf{x}}(k+1 \mid k)] \tag{28}$$

$$P(k \mid N) = P(k \mid k) + C(k)[P(k+1 \mid N) - P(k+1 \mid k)]C(k)' \tag{29}$$

for $k = N-1,\ldots,2,1$. In the above equation,

$$C(k) = P(k \mid k)F(k)'P(k+1 \mid k)^{-1}. \tag{30}$$

Since only one model is used at a time during the forward filtering, there is no need for IMM smoothing here. However, if multiple models are used, then the algorithm proposed in [15] must be used for smoothing with IMM.

The covariance of the smoothed estimates minus the covariance of the unsmoothed estimates must be negative semidefinite. Otherwise, from (29), the smoothed covariances become larger and larger with the backward iteration. If backward transition modifications are performed as the forward transitions, the smoothed variances of the jerk and/or acceleration become larger than their unsmoothed variances. Hence, backward transitions are handled as follows:

1) Wiener process jerk model to free-flight model (white noise jerk model with very small noise and known accelerations) backward transition (at t_3):
   - the smoothed accelerations are set to $[0\ \ 0\ \ -g]$.
   - the covariance is modified using (29) with $F = 12 \times 9$ matrix of zeros except

   $$\begin{aligned} F(1,1) &= F(2,2) = F(3,3) = F(5,4) = F(6,5) \\ &= F(7,6) = F(9,7) = F(10,8) = F(11,9) = 1 \end{aligned} \tag{31}$$

   and $Q = 12 \times 12$ matrix of zeros except

   $$Q(3,3) = Q(7,7) = Q(11,11) = (a_{\max}/2)^2 \tag{32}$$

   $$Q(4,4) = Q(8,8) = Q(12,12) = (j_{\max}/2)^2. \tag{33}$$

   In (29), $P(k+1 \mid k)$ is found using the above F and Q matrices as

   $$P(k+1 \mid k) = FP(k)F' + Q \tag{34}$$

   where k and $k+1$ indicate before and after the transition, respectively.
2) Free-flight model to white noise jerk model backward transition (at t_2):
   - if it is assumed that there is no sudden jump in the acceleration during this transition, then no modification is needed.
   - if a possible jump is assumed in the acceleration, then
     - — the smoothed acceleration estimates are set to the forward filter estimates at the transition time, and
     - — the covariance is modified using (29) with $F = 9 \times 9$ identity matrix and $Q = 9 \times 9$ matrix of zeros except

     $$Q(3,3) = Q(6,6) = Q(9,9) = (a_{\max}/2)^2. \tag{35}$$
3) White noise jerk model to Wiener process jerk model backward transition (at $t_2 - t_d$):
   - the smoothed jerk estimates are set to the forward filter estimates at the transition time, and
   - the covariance is modified using (29) with $F = 9 \times 12$ matrix of zeros except

   $$\begin{aligned} F(1,1) &= F(2,2) = F(3,3) = F(4,5) = F(5,6) \\ &= F(6,7) = F(7,9) = F(8,10) = (9,11) = 1 \end{aligned} \tag{36}$$

   and $Q = 9 \times 9$ matrix of zeros.

Finding the LP estimate using backward prediction from the smoothed estimate at the first measurement time (t_1) is difficult for the following reasons:

- The exact target dynamics, which may have multiple legs with different motion models, are unknown.
- It is possible to have abrupt changes in the acceleration.
- Flying time before the first measurement is unknown.

Uncertainties in the LP estimation can be reduced by using the following additional information:

- The velocity of the target is zero at the launch time.
- The altitude is nearly zero (or a known value based on the local topography) at launch time.

For the LP estimation using the smoothed state $\hat{\mathbf{x}}(1 \mid K)$, the state to be estimated is the state at LP, $\mathbf{x}(0)$, and the "measurement" is $\hat{\mathbf{x}}(1 \mid K)$. Then, the measurement equation for the X coordinate is

$$\underbrace{\begin{bmatrix} \hat{x}(1 \mid K) \\ \hat{\dot{x}}(1 \mid K) \\ \hat{\ddot{x}}(1 \mid K) \\ \hat{\dddot{x}}(1 \mid K) \end{bmatrix}}_{b_x} = \begin{bmatrix} 1 & T & T^2/2 & T^3/6 \\ 0 & 1 & T & T^2/2 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(0) \\ 0 \\ \ddot{x}(0) \\ \dddot{x}(0) \end{bmatrix} + v_x(0) + \omega_x(1) \tag{37}$$

$$= \underbrace{\begin{bmatrix} 1 & T^2/2 & T^3/6 \\ 0 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}}_{A_x} \underbrace{\begin{bmatrix} x(0) \\ \ddot{x}(0) \\ \dddot{x}(0) \end{bmatrix}}_{\mathbf{x}_{\mathrm{LP}_x}} + v_x(0) + \omega_x(1) \tag{38}$$

where $v_x(0)$ is the process noise from launch time to the first measurement time and $\omega_x(1)$ is the "measurement noise" (error in $\hat{\mathbf{x}}(1 \mid K)$), whose covariance is the smoothed estimate's covariance of the X coordinate state at the first measurement time.

A similar equation is used for the Y coordinate. For the Z coordinate,

$$\underbrace{\begin{bmatrix} \hat{z}(1 \mid K) \\ \hat{\dot{z}}(1 \mid K) \\ \hat{\ddot{z}}(1 \mid K) \\ \hat{\dddot{z}}(1 \mid K) \end{bmatrix}}_{b_z} = \begin{bmatrix} 1 & T & T^2/2 & T^3/6 \\ 0 & 1 & T & T^2/2 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \ddot{z}(0) \\ \dddot{z}(0) \end{bmatrix} + v_z(0) + \omega_z(1) \tag{39}$$

$$= \underbrace{\begin{bmatrix} T^2/2 & T^3/6 \\ T & T^2/2 \\ 1 & T \\ 0 & 1 \end{bmatrix}}_{A_z} \underbrace{\begin{bmatrix} \ddot{z}(0) \\ \dddot{z}(0) \end{bmatrix}}_{\mathbf{x}_{\mathrm{LP}_z}} + v_z(0) + \omega_z(1). \tag{40}$$

Since there is no measurement available from the launch time to the first measurement time, the target can be backward-predicted in the X, Y, and Z coordinates separately only if the measurement noises are uncorrelated among coordinates. However, the covariance of the smoothed estimate $\hat{\mathbf{x}}(1 \mid K)$ will have non-zero values for cross-covariance between coordinates. Hence, the above equations for the X, Y, and Z coordinates must be stacked to find the correct least squares estimate.

The least squares estimate of LP is then given by

$$\hat{\mathbf{x}}_{\mathrm{LP}} = (A'\Sigma^{-1}A)^{-1}A'\Sigma^{-1}b \tag{41}$$

where $\mathbf{x}_{\mathrm{LP}} = [\mathbf{x}'_{\mathrm{LP}_x}\ \mathbf{x}'_{\mathrm{LP}_y}\ \mathbf{x}'_{\mathrm{LP}_z}]'$, $A = \mathrm{diag}(A_x, A_y, A_z)$, $b = [b'_x\ b'_y\ b'_z]'$, and Σ is the sum of the covariance of the smoothed estimate at the first measurement time and the covariance Q_0 of the process noise $v(0)$.

The covariance of the least squares estimate is given by

$$\mathrm{cov}(\hat{\mathbf{x}}_{\mathrm{LP}}) = (A'\Sigma^{-1}A)^{-1}. \tag{42}$$

In the above covariance calculation, launch time is assumed to be known. However, there is uncertainty in T, hence in A.

To include the uncertainty in T, T is stacked with the state vector to yield the augmented state $\mathbf{y}_{\mathrm{LP}} = [\mathbf{x}'_{\mathrm{LP}}\ T]'$. Since this is a nonlinear least squares problem, the ILS approach, as described in section 4.2, is used to estimate the launch point. The linearized measurement matrix becomes

$$\bar{A} = \begin{bmatrix}
1 & T^2/2 & T^3/6 & 0 & 0 & 0 & 0 & 0 & \ddot{x}(0)T + \dddot{x}(0)T^2/2 \\
0 & T & T^2/2 & 0 & 0 & 0 & 0 & 0 & \ddot{x}(0) + \dddot{x}(0)T \\
0 & 1 & T & 0 & 0 & 0 & 0 & 0 & \dddot{x}(0) \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & T^2/2 & T^3/6 & 0 & 0 & \ddot{y}(0)T + \dddot{y}(0)T^2/2 \\
0 & 0 & 0 & 0 & T & T^2/2 & 0 & 0 & \ddot{y}(0) + \dddot{y}(0)T \\
0 & 0 & 0 & 0 & 1 & T & 0 & 0 & \dddot{y}(0) \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & T^2/2 & T^3/6 & \ddot{z}(0)T + \dddot{z}(0)T^2/2 \\
0 & 0 & 0 & 0 & 0 & 0 & T & T^2/2 & \ddot{z}(0) + \dddot{z}(0)T \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & T & \dddot{z}(0) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}. \tag{43}$$

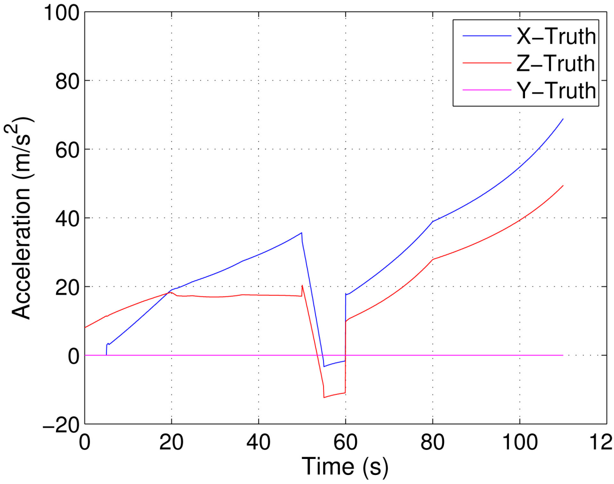


Fig. 2. Boost phase acceleration in each coordinate.

The estimate at iteration $j+1$ is given by

$$\hat{\mathbf{y}}_{\rm LP}^{j+1} = \hat{\mathbf{y}}_{\rm LP}^{j} + (\bar{A}'\Sigma^{-1}\bar{A})^{-1}\bar{A}'\Sigma^{-1}[b - \bar{A}\hat{\mathbf{y}}_{\rm LP}^{j}]. \qquad (44)$$

An initial estimate $\hat{\mathbf{y}}_{\rm LP}^{0}$ for the ILS estimator is obtained using (41) with an assumed T.

The covariance of the iterative least squares estimate with uncertainty in the launch time is

$$\mathrm{cov}(\hat{\mathbf{y}}_{\rm LP}) = (\bar{A}'\Sigma^{-1}\bar{A})^{-1}. \qquad (45)$$

Without any additional constraints, the solution of (44) might converge to a wrong estimate if the initial guess for the launch time is not close to the actual value. The solution can be improved by adding additional constraints for the accelerations

$$\mathrm{sign}(\hat{\ddot{x}}(1 \mid K)) = \mathrm{sign}(\dot{x}(0)) \qquad (46)$$

$$\mathrm{sign}(\hat{\ddot{y}}(1 \mid K)) = \mathrm{sign}(\dot{y}(0)) \qquad (47)$$

$$\mathrm{sign}(\hat{\ddot{z}}(1 \mid K)) = \mathrm{sign}(\dot{z}(0)). \qquad (48)$$

The above constraints say that the direction of the acceleration must be same as the direction of the velocity component of b. A constrained nonlinear least squares algorithm is required to solve the above problem [9]. MATLAB “fmincon” function can be used to solve the above problem.

4.5. Posterior Cramer-Rao Lower Bound

The PCRLB, which is defined as the inverse, $J(k)^{-1}$, of the Fisher Information Matrix (FIM), gives a lower bound on the error covariance [20]

$$E\{[\hat{\mathbf{x}}(k) - \mathbf{x}(k)][\hat{\mathbf{x}}(k) - \mathbf{x}(k)]'\} \geq J(k)^{-1} \qquad (49)$$

where E denotes expectation over ($\mathbf{x}(k)$, $Z(k) = [z(1), z(2), \ldots, z(k)]$).

For a system with a linear dynamic model and a nonlinear measurement model without measurement origin uncertainty, $J(k+1)$ can be written as [8]

$$J(k+1) = [Q(k) + F(k)J(k)^{-1}F(k)']^{-1} + \sum_{i=1}^{n} E(H^i(k)'R^i(k)^{-1}H^i(k)) \qquad (50)$$

with the (m,n)th element of matrix $H_i(k)$ being given by

$$[H^i(k)](m,n) = \frac{\partial[h^i(k)](m)}{\partial\mathbf{x}(k)(n)}. \qquad (51)$$

The PCRLB for smoothing without measurement origin uncertainty is the same as (29) [19].

During model transitions, the PCRLB can be modified in the same manner as the covariance modifications discussed in the forward filtering and smoothing sections.

5. SIMULATION RESULTS

The simulation settings are as follows:

- Target launch location at $T_0 = 0$ is 30°N latitude, 45°E longitude and 0 m altitude with heading east.
- First measurement is received 20 s after launch.
- The sampling interval is 0.2 s.
- The measurement error standard deviation (azimuth and elevation) is 10 μrad.
- Two IR sensors located on geostationary satellites.
  —Satellite 1 located at time T_0 is: 0° latitude, 5°W longitude and 37000 km altitude.
  —Satellite 2 located at time T_0 is: 0° latitude, 60°E longitude and 37000 km altitude.
- Measurements are synchronized between the two satellites.

The target trajectory is based on real data [4] and measurements are generated using simulation. The net acceleration magnitude of the target is shown in Fig. 1. Acceleration in each coordinate is shown in Fig. 2. During the free-flight motion, the acceleration is assumed to be $[0, 0, -10]$ m/s^2, however the actual values of the acceleration in the X-direction vary from -3.3 to -1.7 m/s^2 and in the Z-direction vary from -12.3 to -11 m/s^2. The atmospheric drag, which acts opposite to the target velocity vector, is the reason for the mismatch between the assumed and the actual accelerations. There is no mismatch in the acceleration in the Y-direction, since the velocity, hence drag, in that direction is zero. The first stage boost decay time t_d is assumed to be 5 s. A possible jump was assumed in the acceleration during the model transition at t_2. A constrained Kalman Filter is used to impose a minimum acceleration with $\ddot{x}_{\min} = -2$ m/s^2, $\ddot{y}_{\min} = -2$ m/s^2 and $\ddot{z}_{\min} = -12$ m/s^2.

5.1. EOB Estimate

The ground truth and estimates of the position of the targets from a sample run are shown in Fig. 3. Fig. 4 shows a magnified plot of Fig. 3 around the first measurement time. The velocity, acceleration, and

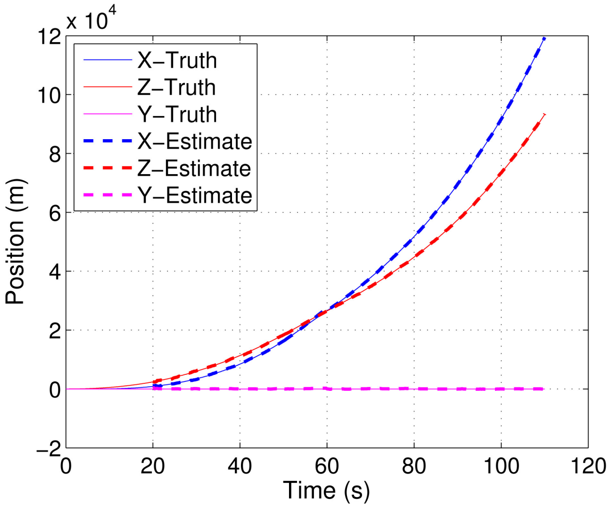


Fig. 3. Position estimates from a sample run.

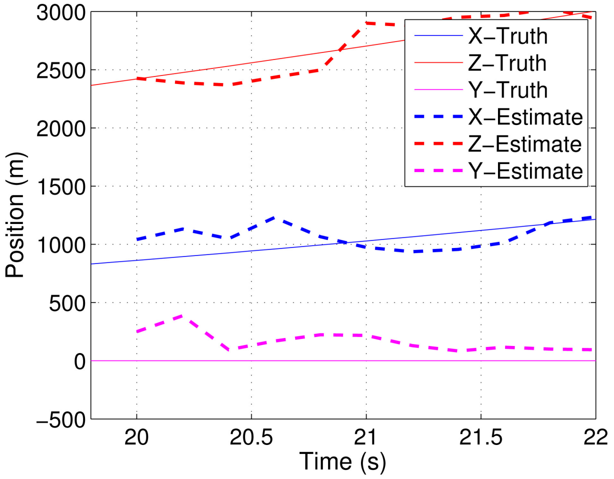


Fig. 4. Magnified plot of position estimates around the first measurement time.

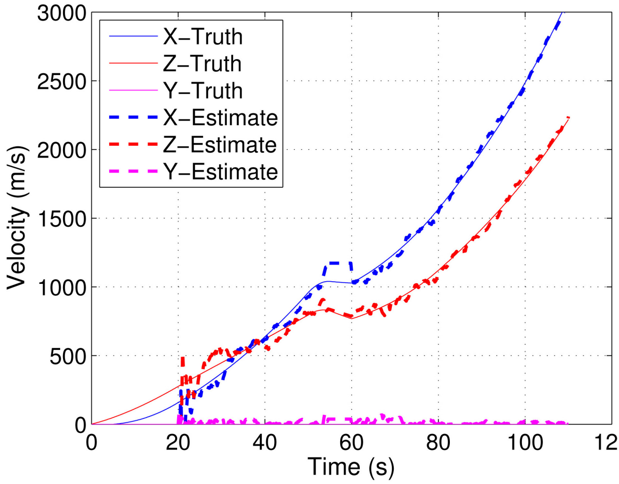


Fig. 5. Velocity estimates from a sample run.

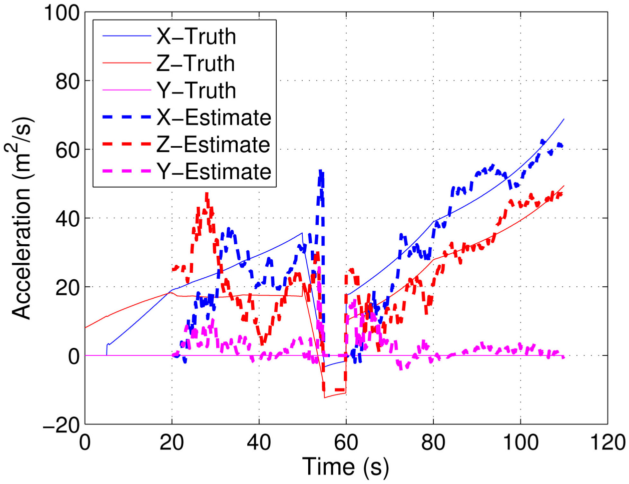


Fig. 6. Acceleration estimates from a sample run.

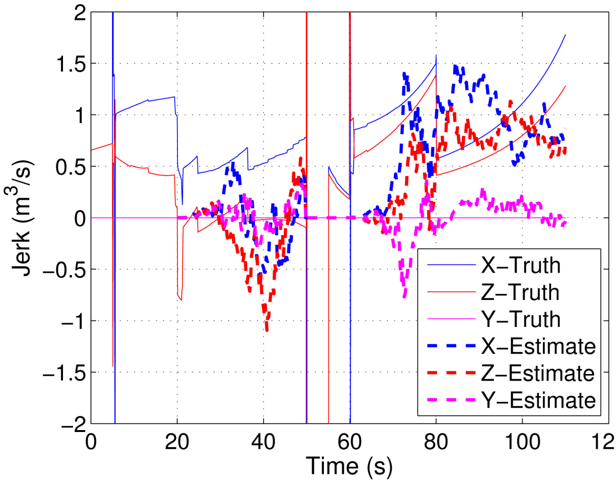


Fig. 7. Jerk estimates from a sample run.

jerk estimates from a sample run are shown in Figs. 5, 6, and 7, respectively. Due to insufficient measurement accuracy and short sampling duration, the acceleration reduction of the target during the time 50 s to 55 s is not estimated accurately. Jerk estimates are also not accurate. Even though jerk estimates are not accurate, the position, velocity and acceleration estimates are reasonably accurate.

The Root Mean Square Error (RMSE) values of position estimates calculated from 100 Monte-Carlo runs are shown in Fig. 8. When all the measurements are used, the RMSE at the EOB is around 125 m. Position RMSE is continuously increasing from time 55 s to 60 s because of absence of measurements and mismatch between assumed and actual acceleration. The position RMSE with the measurements of the second stage of the boost phase of the target is also shown in Fig. 8. The corresponding velocity RMSE is shown in Fig. 9. From these figures, it can be noticed that due to the high uncertainty in the acceleration and jerk at the start of second stage of the boost phase and high process noise, information from 20 s to 50 s did not give any improvement to the RMSE of the EOB state estimate. Position and velocity PCRLBs at EOB with measurements of different start times are shown in

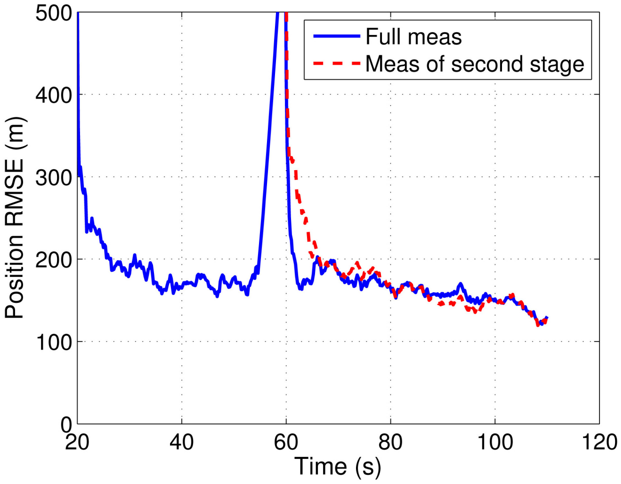


Fig. 8. EOB RMSE of position estimates with all measurements versus only measurements after 60 seconds.

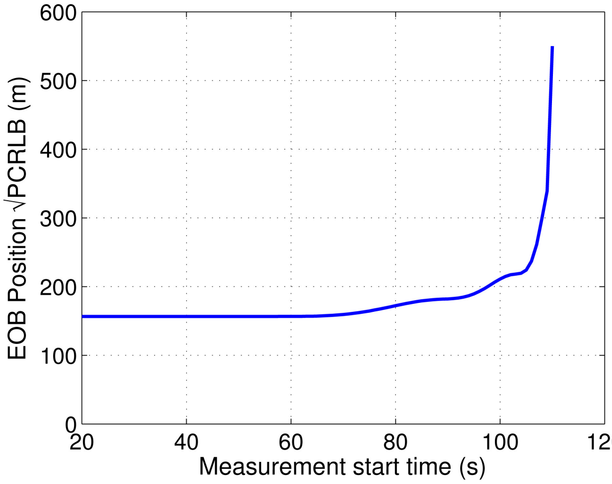


Fig. 10. Square root of position PCRLB of EOB state estimate with different measurement start time.

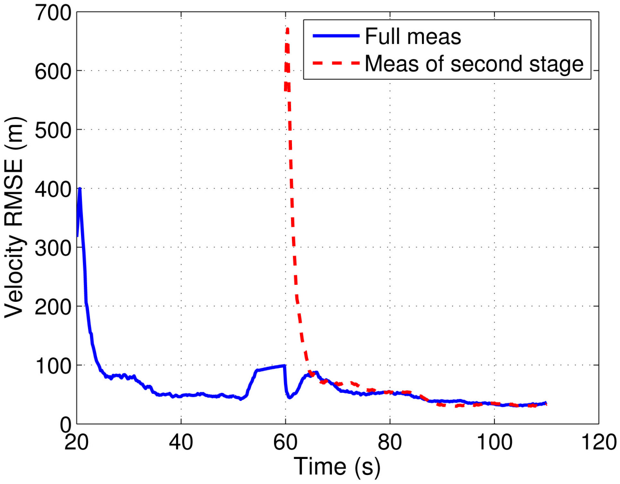


Fig. 9. EOB RMSE of velocity estimates with all measurements versus only measurements after 60 seconds.

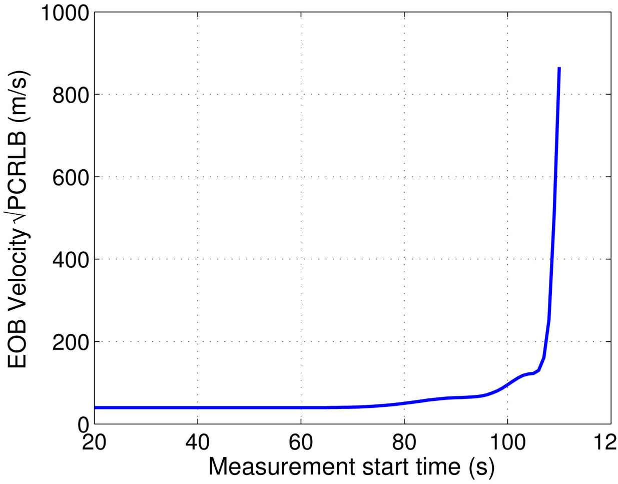


Fig. 11. Square root of velocity PCRLB of EOB state estimate with different measurement start time.

Figs. 10 and 11, respectively. From these two figures, it can be noticed that the measurements before 65 s do not add significant information to EOB state estimate.

5.2. LP Estimate

As expected, smoothed velocity, acceleration and jerk estimates are better than the filtered estimates without smoothing, as shown in Figs. 12, 13, and 14. Position and velocity RMSEs are also significantly improved by smoothing, as shown in Figs. 15 and 17. Since the velocity at the LP is assumed to be zero, there was no need to estimate it. As a result, the velocity RMSE is close to zero. The position RMSE at LP is around 500 m, which is a reasonable value for launch point estimation problem. The position RMSE of the launch point estimate is much higher than the estimate at the first measurement time (20 s), since no measurements were available from 0 s to 20 s. Figs. 16 and 18 show the PCRLB values corresponding to position and velocity RMSEs, respectively. The PCRLB and RMSE values are similar except at LP. The lower PCRLB compared to the RMSE at LP could be due to model mismatch.

The position and velocity RMSE of the smoothed estimates with measurements from first stage of boost phase of the target are shown in Figs. 19 and 20, respectively. From these figures it can be noticed that the measurements from the second stage of the boost phase do not contribute significantly to the LP estimation due to the sudden changes in acceleration and jerk during stage transition and the high process noises. Position PCRLBs of LP estimate using measurements with different end times are shown in Fig. 21. From this figure it can be noticed that the measurements after 50 s do not add significant information to LP estimate.

The comparison of the Wiener process jerk model with exponentially autocorrelated acceleration model is shown in Figs. 22 and 23. In this comparison, the following values are used for α: $\alpha_x = -0.02$, $\alpha_y = -0.02$, $\alpha_z = -0.01$. Except for the LP estimate, the RMSEs of both models are almost at the same level, since both

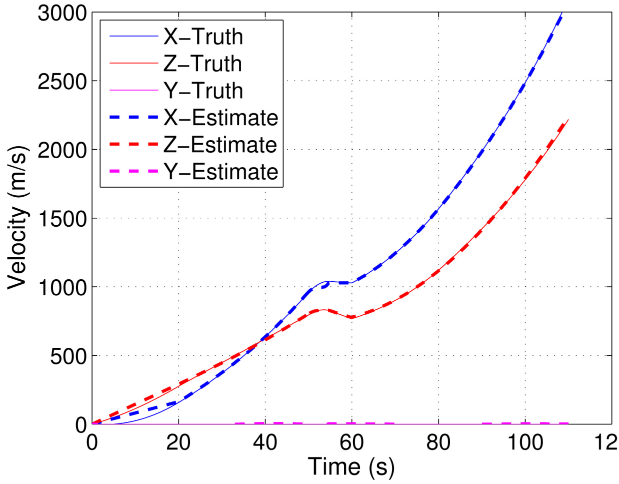


Fig. 12. Smoothed estimates of velocity from a sample run.

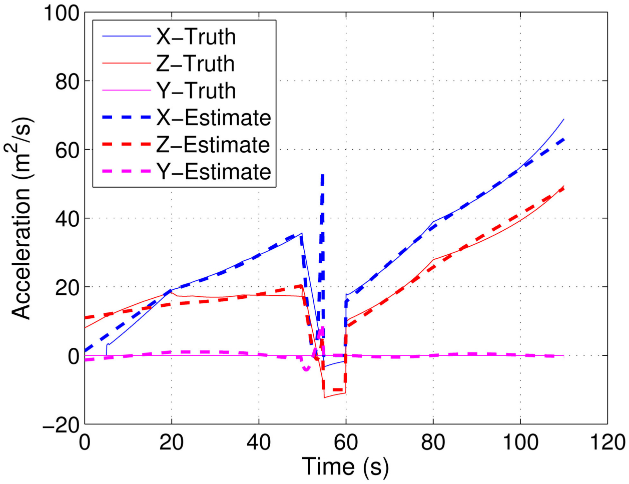


Fig. 13. Smoothed estimates of acceleration from a sample run.

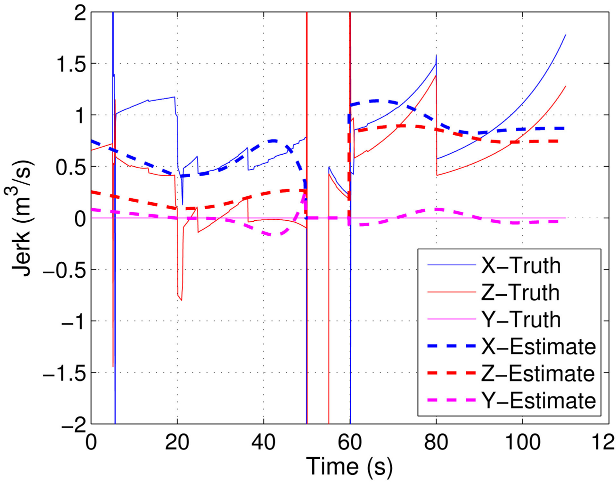


Fig. 14. Smoothed estimates of jerk from a sample run.

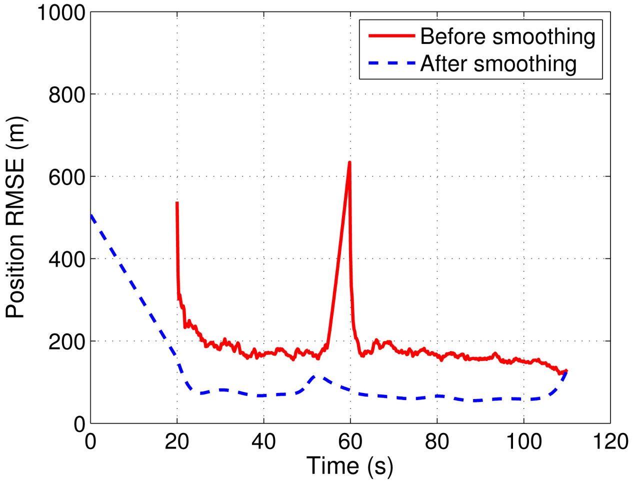


Fig. 15. Position RMSE.

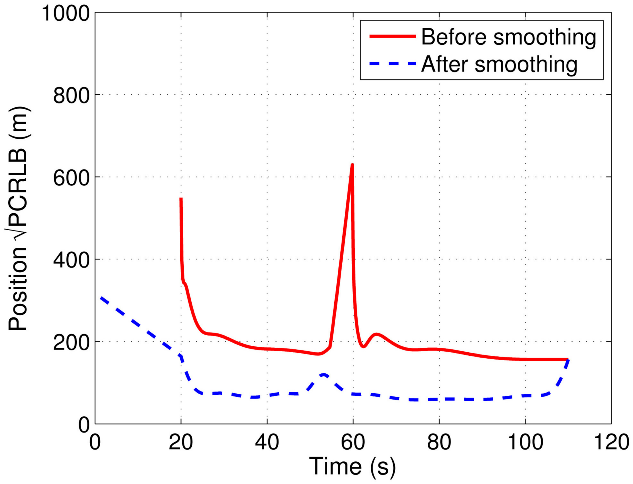


Fig. 16. Square root of position PCRLB.

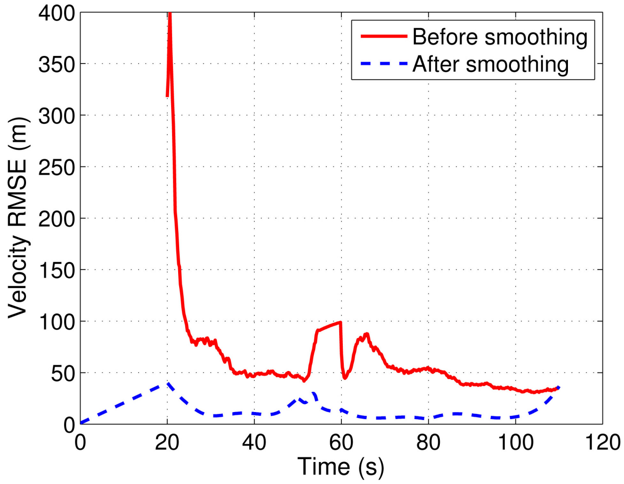


Fig. 17. Velocity RMSE.

models have almost the same error in approximating the actual model.

The RMSE, bias, and Standard Deviation (STD) of the LP estimate with the Wiener process jerk model and the exponentially autocorrelated acceleration model are shown in the columns two to five of Table I. The value of covariance Q_0 of the process noise $v(0)$ is set using (6) with PSD $q_m = 0.01$ m^{2}s^{-7}. For the Wiener process

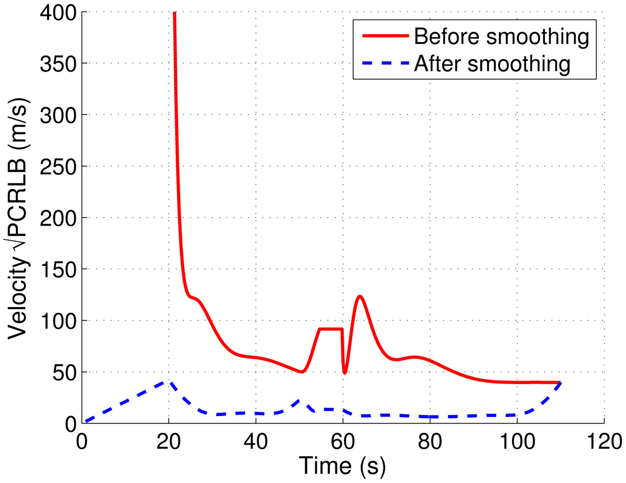


Fig. 18. Square root of velocity PCRLB.

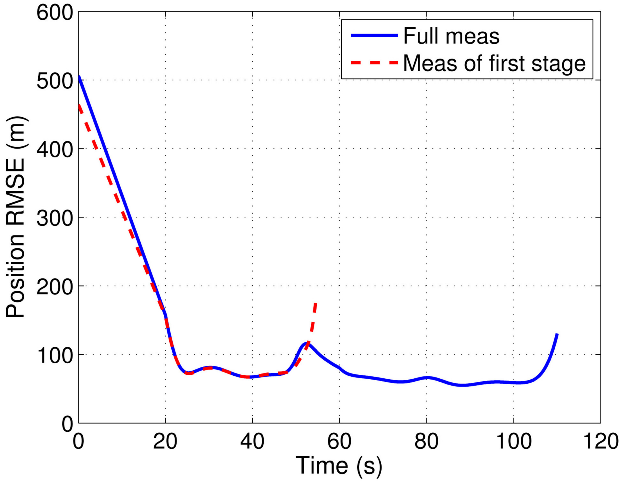


Fig. 19. RMSE of smoothed position estimates with all measurements versus only measurements of first stage of boost phase.

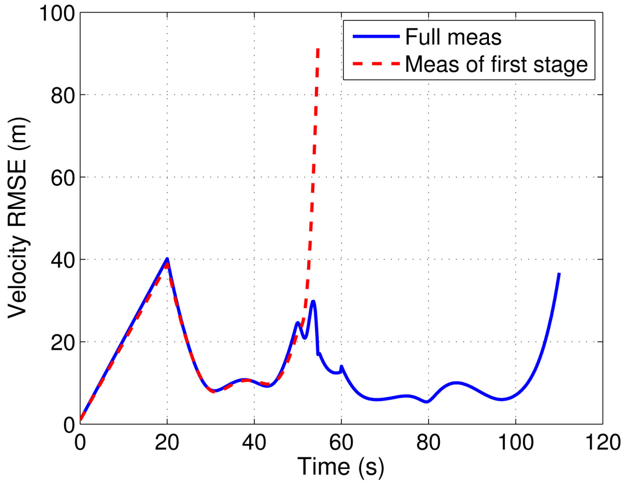


Fig. 20. RMSE of smoothed velocity estimates with all measurements versus only measurements of first stage of boost phase.

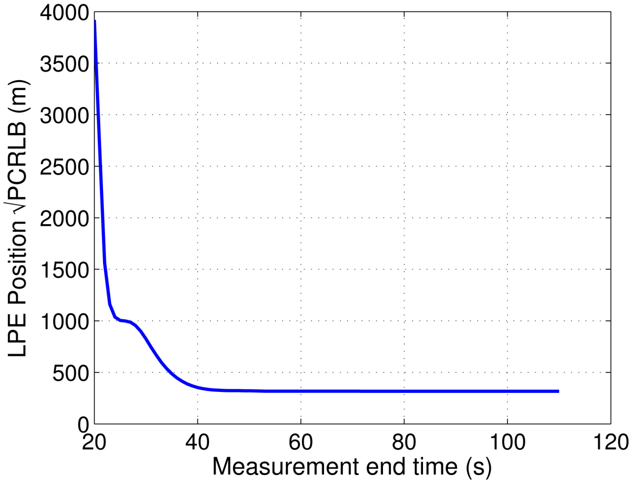


Fig. 21. Square root of position PCRLB of LP estimate with different measurement end time.

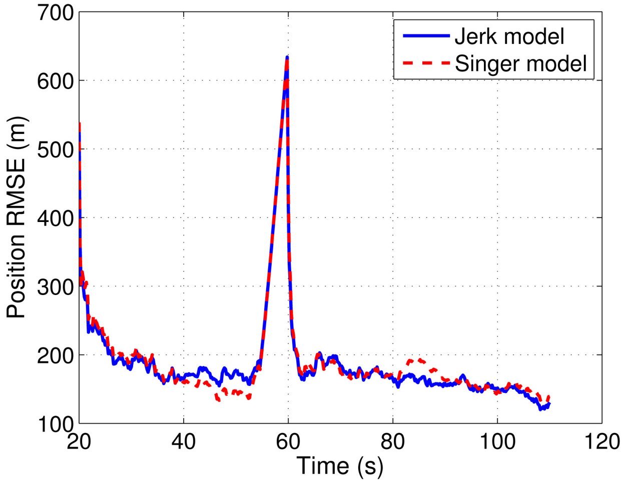


Fig. 22. RMSE of position estimates with Wiener process jerk model versus exponentially autocorrelated acceleration model.

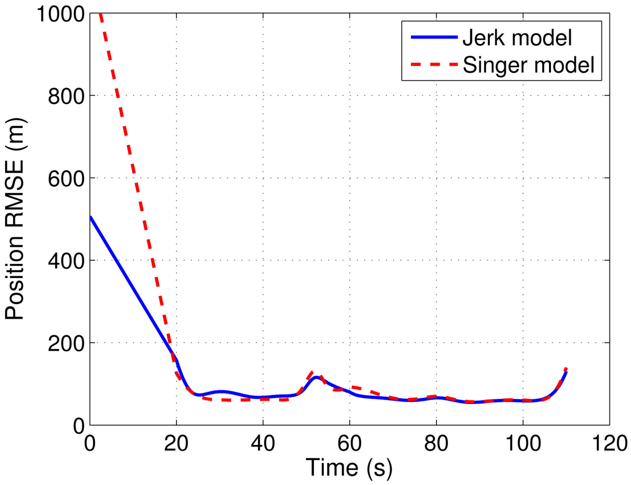


Fig. 23. RMSE of smoothed position estimates with Wiener process jerk model versus exponentially autocorrelated acceleration model.

TABLE I
Bias, RMSE and Standard Deviation of LP Estimate Calculated from 100 Monte-Carlo runs with 20 s Delay Before First Measurement

	CWPJ Model and Known Launch Time		Exponentially Autocorrelated Acceleration Model and Known Launch Time		CWPJ Model and Estimated Launch Time		CWPJ Model and Wrong Launch Time (assumed delay 25 s)	
	x (m)	y (m)	x (m)	y (m)	x (m)	y (m)	x (m)	y (m)
RMSE	376.97	185.50	855.98	112.53	368.72	130.71	381.53	273.07
Bias	−295.70	28.85	−846.41	6.10	−309.12	12.87	−170.46	52.49
STD	235.00	184.16	128.23	112.93	202.01	130.73	343.05	269.33

jerk model, the RMSE value in the X-direction is significantly larger than the STD due to the bias in the estimate. The bias in the estimate of the X coordinate is due to sudden change in the acceleration after 5 s from the launch (see Fig. 2), which introduces a model mismatch. For the exponentially autocorrelated acceleration model, the smoothed state's error standard deviation is reasonably small, however the RMSE values are large due to the large biases. These biases are due to the mismatch between the actual and used α values. It is very difficult to estimate a value for α in the time duration 0 to 20 seconds without any measurements.

So far, the launch time has been assumed known (i.e., the launch time is assumed first measurement time minus 20 s). When the launch time is unknown, the launch point estimation results obtained using MATLAB "fmincon" with constraints (46)–(46) are shown in the columns six and seven of Table I. The average of the launch time estimate $\hat{T}$ is 18.41 s. In order to analyze the effect of error in the launch time, smoothing is performed by assuming that the launch time is the first measurement time minus 25 s using (41). The corresponding RMSE, bias and STD of the LP estimate are shown in the last two columns of Table I. From Table I, it can be noticed that a small error in the launch time does not increase the position RMSE significantly. Even though the error in the launch time might affect the initial acceleration and jerk estimates, the launch time and the initial acceleration and jerk do not appear to be as vital as the launch position estimate.

6. CONCLUSION

In this paper, the launch point estimation of a ballistic target using the measurements from satellite-borne passive sensors during the boost phase of the target is considered. The target is assumed to have a two-stage boost phase with a free-flight phase between the two stages. A profile-free method with an adaptive model selection relying on the free-flight duration (assumed to be observed) is proposed for end-of-burnout and launch point estimation of ballistic targets. Measurements are assumed available only staring at a certain time after the launch. The launch point is estimated using smoothing followed by a least squares estimator. Prior information on the launch state such as zero initial velocity and zero (or known) altitude at launch are used to improve the estimate of the launch point.

From the simulation results as well as the PCRLB, it was observed that the launch point estimates were not improved by using measurements from the second stage of the boost phase. Hence, if the launch point estimate is the only item of interest, then there is no need for complicated filters like IMM to detect the maneuvers that occur after the time at which the accuracy of launch point estimate saturates. Also, the simulation results suggested that the exponentially autocorrelated acceleration model is not a good choice for launch point estimation and the Wiener process jerk model is the better.

APPENDIX. NOMENCLATURE

$E\{\cdot\}$	Expectation operator
F	State transition matrix
γ	Elevation
Q	Process noise covariance
h	Nonlinear measurement function
$\bar{H}$	Jacobian of nonlinear measurement function
$\bar{\bar{H}}$	Jacobian of stacked nonlinear measurement functions
J	Fisher information matrix
P	Covariance of the state estimate
R	Measurement error covariance
T	Sampling period
θ	Azimuth
$\mathbf{x}$	State vector
$\hat{\mathbf{x}}$	Estimate of state vector $\mathbf{x}$
$\mathbf{z}$	Measurement vector

REFERENCES

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

[2] Y. Bar-Shalom, X. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation.
New York: Wiley, 2001.

[3] A. Benavoij, L. Chisci, and A. Farina
Tracking of a ballistic missile with a-priori information.
IEEE Transactions on Aerospace and Electronic Systems, **43**, 3 (July 2007), 1000–1016.

[4] W. D. Blair, J. Lawton, and G. Martell
Private communication from NSWC Dahlgren Division, VA, 1995, 1997.

[5] N. J. Danis
Space-based tactical ballistic missile launch parameter estimation.
IEEE Transactions on Aerospace and Electronic Systems, **29**, 2 (Apr. 1993), 412–424.

[6] A. Farina, L. Timmoneri, and D. Vigilante
Classification and launch-impact point prediction of ballistic target via multiple model maximum likelihood estimator (MM-MLE).
In Proceedings of the IEEE Conference on Radar, Verona, NY, Apr. 2006.

[7] W. J. Farrell
Interacting multiple model filter for tactical ballistic missile tracking.
IEEE Transactions on Aerospace and Electronic Systems, **44**, 2 (Apr. 2008), 418–426.

[8] M. L. Hernandez, T. Kirubarajan, and Y. Bar-Shalom
Multisensor resource deployment using posterior Cramér-Rao bounds.
IEEE Transactions on Aerospace and Electronic Systems, **40**, 2 (Apr. 2004), 399–416.

[9] J. N. Holt and R. Fletcher
An algorithm for constrained nonlinear least squares.
Journal of the Institute of Mathematics and Its Applications, **23**, 4 (1979), 449–463.

[10] R. G. Hutchins and P. E. Pace
Studies in trajectory tracking and launch point determination for ballistic missile defense.
In Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, 6236-0Y, Orlando, FL, Apr. 2006.

[11] X. R. Li and V. P. Jilkov
A survey of maneuvering target tracking—Part II: ballistic target models.
Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, San Diego, CA, Aug. 2001.

[12] Y. Li, T. Kirubarajan, M. Yeddanapudi and Y. Bar-Shalom
Trajectory and launch point estimation for ballistic missiles based on boost-phase LOS measurements.
In Proceedings of the IEEE Aerospace Conference, Snowmass, CO, Mar. 1999, 425–442.

[13] L. N. Lillard, H. E. Evans, and J. J. Spaulding
Minimum variance missile launch and impact estimation by fusing observations from multiple sensors.
In Proceedings of the IEEE Aerospace Conference, 3, Aspen, CO, Feb. 1997, 309–320.

[14] V. P. S. Naidu, G. Girija, and J. R. Raol
Estimation of launch and impact points of a flight trajectory using U-D Kalman filter/smoother.
Defence Science Journal, **56**, 4 (Oct. 2006), 451–463.

[15] N. Nandakumaran, R. Tharmarasa, T. Lang, M. McDonald, and T. Kirubarajan
Interacting multiple model forward filtering and backward smoothing for maneuvering target tracking.
In Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, 7445-03, San Diego, CA, Aug. 2009.

[16] E. Nelson, M. Pachter, and S. Musick
Projectile launch point estimation from radar measurements.
In Proceedings of the American Control Conference, 2, Portland, OR, June 2005, 1275–1282.

[17] R. W. Osborne, III and Y. Bar-Shalom
Statistical efficiency of composite position measurements from passive sensors.
In Proceedings of SPIE Conference on Signal Processing, Sensor Fusion and Target Recognition, 8050-07, Orlando, FL, Apr. 2011.

[18] P. W. Richards
Constrained Kalman filtering using pseudo-measurements.
In Proceedings of the IEE Colloquium on Algorithms for Target Tracking, London, UK, May 1995.

[19] M. Simandl, J. Kralovec, and P. Tichavsky
Filtering, predictive, and smoothing Cramer-Rao bounds for discrete-time nonlinear dynamic systems.
Automatica, **37** (2001), 1703–1716.

[20] H. Van Trees
Detection, Estimation and Modulation Theory, Vol. I.
New York: Wiley, 1968.

[21] J. R. Van Zandt
Boost phase tracking with an unscented filter.
In Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, 4728, Orlando, FL, Apr. 2002, 263–274.

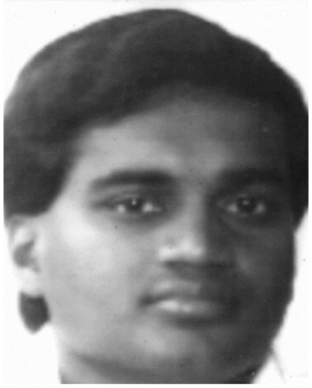
[22] P. Zarchan
Boost phase filtering options: is simpler better?
Submitted to *AIAA J-G&C*, 2010.

Ratnsingham Tharmarasa was born in Sri Lanka in 1975. He received the B.Sc.Eng. degree in electronic and telecommunication engineering from University of Moratuwa, Sri Lanka in 2001, and the M.A.Sc. and Ph.D. degrees in electrical engineering from McMaster University, Canada in 2003 and 2007, respectively.

From 2001 to 2002 he was an instructor in electronic and telecommunication engineering at the University of Moratuwa, Sri Lanka. During 2002–2007 he was a graduate student/research assistant in ECE department at the McMaster University, Canada. Currently he is working as a research associate in the Electrical and Computer Engineering Department at McMaster University, Canada. His research interests include target tracking, information fusion and sensor resource management.

Thiagalingam Kirubarajan (S'95—M'98—SM'03) was born in Sri Lanka in 1969. He received the B.A. and M.A. degrees in electrical and information engineering from Cambridge University, England, in 1991 and 1993, and the M.S. and Ph.D. degrees in electrical engineering from the University of Connecticut, Storrs, in 1995 and 1998, respectively.

Currently, he is a professor in the Electrical and Computer Engineering Department at McMaster University, Hamilton, Ontario. He is also serving as an Adjunct Assistant Professor and Associate Director of the Estimation and Signal Processing Research Laboratory at the University of Connecticut. His research interests are in estimation, target tracking, multisource information fusion, sensor resource management, signal detection and fault diagnosis. His research activities at McMaster University and at the University of Connecticut are supported by U.S. Missile Defense Agency, U.S. Office of Naval Research, NASA, Qualtech Systems, Inc., Raytheon Canada Ltd. and Defense Research Development Canada, Ottawa. In September 2001, Dr. Kirubarajan served in a DARPA expert panel on unattended surveillance, homeland defense and counterterrorism. He has also served as a consultant in these areas to a number of companies, including Motorola Corporation, Northrop-Grumman Corporation, Pacific-Sierra Research Corporation, Lockhead Martin Corporation, Qualtech Systems, Inc., Orincon Corporation and BAE systems. He has worked on the development of a number of engineering software programs, including BEARDAT for target localization from bearing and frequency measurements in clutter, FUSEDAT for fusion of multisensor data for tracking. He has also worked with Qualtech Systems, Inc., to develop an advanced fault diagnosis engine.



Dr. Kirubarajan has published about 100 articles in areas of his research interests, in addition to one book on estimation, tracking and navigation and two edited volumes. He is a recipient of Ontario Premier's Research Excellence Award (2002).

Nandakumaran Nadarajah (S'05—M'10) was born in Sri Lanka in 1976. He received the B.Sc.Eng. degree in electrical and electronic engineering from University of Peradeniya, Peradeniya, Sri Lanka, in 2001, and the M.A.Sc. and Ph.D. degrees in electrical engineering from McMaster University, Canada, in 2005 and 2009, respectively.

From 2002 to 2003 he was an assistant lecturer in Electrical and Electronic Engineering at the University of Peradeniya. From 2003 to 2009 he was a graduate student/research assistant in Electrical and Computer Engineering Department at McMaster University, Canada. Currently, he is working as a postdoctoral research fellow at Curtin University, Perth, Australia. His research interests are in signal processing, target tracking, data fusion, GNSS navigation and attitude determination.

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.

Thayananthan Thayaparan earned a B.Sc. (Hons.) in physics at the University of Jaffna, Srilanka in 1987, an M.Sc. in physics at the University of Oslo, Norway in 1991, and a Ph.D. in atmospheric physics at the University of Western Ontario, Canada in 1996.

From 1996 to 1997, he was employed as a postdoctoral fellow at the University of Western Ontario. In 1997, he joined the Defence Research and Development Canada–Ottawa, Department of National Defence, Canada, as a defence scientist. His research interests include advanced radar signal and image processing methodologies and techniques against SAR/ISAR and HFSWR problems such as detection, classification, recognition, and identification. His current research includes computational synthetic aperture radar imaging algorithms, time-frequency analysis for radar detection, imaging and signal analysis, radar micro-Doppler analysis, and concealed weapon detection using radars.

Dr. Thayaparan is currently serving in the Editorial Board of IET Signal Processing. He has authored or coauthored over 170 publications in journals, proceedings, and internal distribution reports. He is a fellow of the IET (previously IEE). He has been appointed as an adjunct professor at McMaster University.

Predetection Fusion in Large Sensor Networks with Unknown Target Locations

RAMONA GEORGESCU
PETER WILLETT
STEFANO MARANO
VINCENZO MATTA

Fusion of multisensor data can improve target probability of detection but suffers from a potentially increased false alarm rate. The optimal sensor decision rule in the case of multiple sensor systems and known target location is of course a likelihood ratio test. This approach, however, is not applicable to many practical scenarios, such as sonar, in which the location of the target is not known and hence the alternative hypothesis becomes composite. Therefore, we propose predetection fusion and highlight its application to a variety of multitarget multisensor trackers. Additionally, the algorithm is motivated by the need for an efficient way to process the volume of data from large sensor networks that consist of low quality sensors. We thus propose predetection fusion as a contact sifting procedure followed by an Expectation Maximization step that refines the location of the estimated detections. Results are provided on a synthetic dataset and on a challenging realistic multistatic sonar dataset. The performance of predetection fusion is compared against the performance of the optimal multi-hypothesis GLRT approach.

Manuscript received October 20, 2010; revised November 28, 2011; released for publication December 11, 2011.

Refereeing of this contribution was handled by Pramod Varshney.

This work was supported by the Office of Naval Research under contract N00014-10-10412.

Authors' addresses: R. Georgescu and P. Willett, Electrical and Computer Engineering Department, University of Connecticut, Storrs, CT 06269, E-mail: ({ramona, willett}@engr.uconn.edu); S. Marano and V. Matta, DIIIE, University of Salerno, via Ponte don Melillo I-84084, Fisciano (SA), Italy, E-mail: ({marano, vmatta}@unisa.it).

1557-6418/12/$17.00 © 2012 JAIF

1. MOTIVATION

Our motivation to devise this predetection fusion algorithm was twofold: one reason is the inapplicability of traditional *decentralized detection* to a realistic scenario in which target locations are unknown, while the other was the need for an efficient way to process the data generated in a large network of low quality sensors. As our ultimate goal is tracking, we will end this section by pointing out the benefits of predetection fusion to that.

1.1. Detection Fusion

In the 1980s and the 1990s, many papers appeared that dealt with optimal detection for multiple sensor systems, among them [5], [25], [27], [28], [29] and [30]. The problem under consideration was a binary detection problem with N sensors, and the target location was implicitly assumed known. The purpose of the detector was to optimally discriminate between the two simple hypotheses:

$$H_0: \quad \text{noise only},$$
$$H_1: \quad \text{target present} + \text{noise}.$$

In a practical case, such as sonar/radar detections, the location of the target is not known and the alternative hypothesis becomes composite. Hence, these approaches, generally based on likelihood ratio tests, cannot be directly applied. We here propose a practical implementation, which we will refer to as predetection fusion.

1.2. Large Networks of Low Quality Sensors

Data fusion in large sensor networks is expected to provide better target tracking capability in terms of increased area coverage, expanded geometric diversity, increased target hold, robustness to sensor loss and jamming, improved localization, and gains in probability of detection [10]. A possible drawback is an increased false alarm rate after an unwary fusion step.[1] Predetection fusion is a data fusion technique that attempts to maintain the target probability of detection while reducing the false alarm rate.

An example of such a sensor network is a multistatic sonar system, which consists of multiple sonar sources and receivers distributed over the surveillance area [10]. In recent years, interest has shifted towards deploying large sensor networks that consist of many but cheap, low quality sensors. The difficult Metron multistatic sonar dataset [23] is representative of such a setup.

In one scenario of the Metron dataset, there are 25 stationary sensors located as in Fig. 1. All the sensors are receivers with the exception of four which are colocated source/receiver units. The probability of detection

[1] What we mean is that with *direct* data fusion (i.e., pooling all available measurements), a target seen once may be repeatedly observed as each sensor offers its own perspective. While this may offer some advantage in terms of confidence and drill-down of localization, the pooled false alarm density is multiplied by the sensor count.

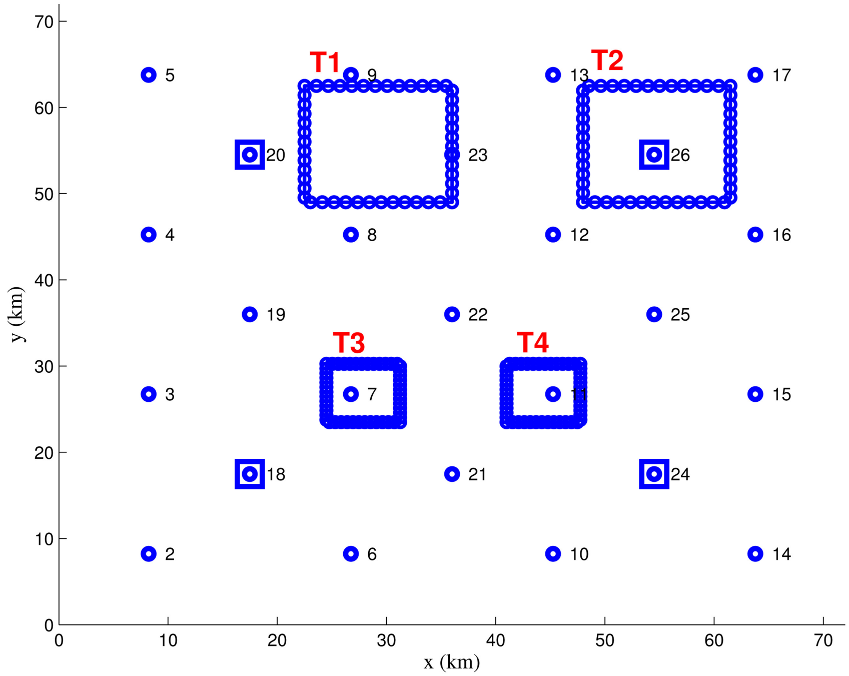


Fig. 1. Setup in the Metron dataset (4 sources S1–S4, 25 receivers RX2–RX26, 4 targets with square trajectories T1–T4).

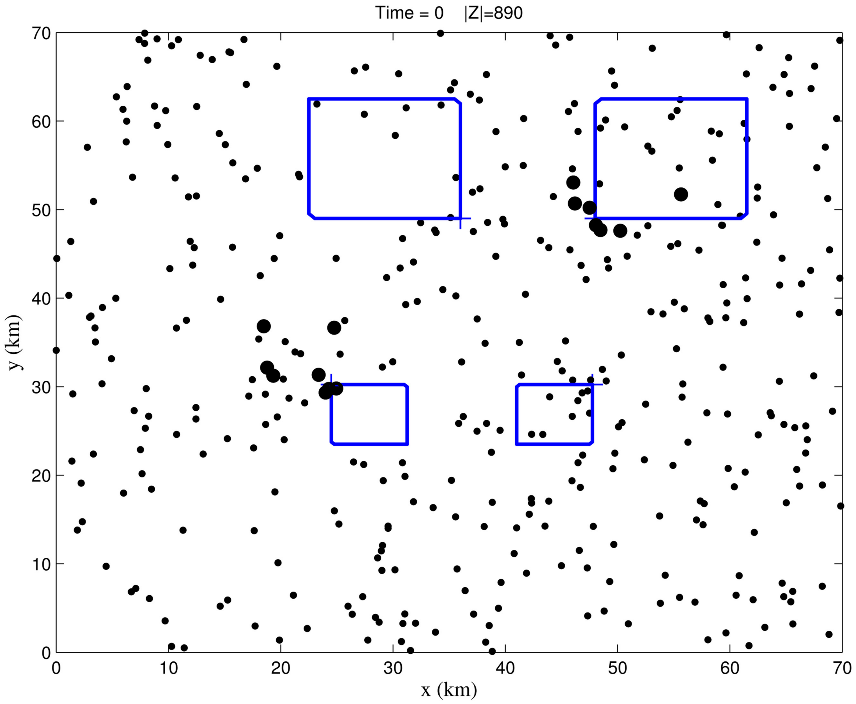


Fig. 2. Raw measurements collected from all sensors in a single scan (target originated measurements are emphasized).

is poor, on average $P_D = 0.12$ per sensor per scan. Target 1 starts in the lower right corner of its square trajectory and moves clockwise, target 2 starts in the lower left corner and moves counterclockwise, target 3 starts in the upper left corner and moves counterclockwise and target 4 starts in the upper right corner and moves clockwise. The contacts generated by targets 2 and 3 have been tagged and the ground truth is available [23]. Some further description of the Metron data is in [12], [13].

The high difficulty of the dataset is due to the extremely large number of contacts per scan and the low quality of the measurements. Fig. 2 shows the first scan of data plotted in Cartesian coordinates: there are 890 contacts out of which only 15 originate from a target, already a major challenge to any tracking paradigm. Moreover, Fig. 3 shows that the data is of very low quality: the 1-sigma covariance ellipses are very elongated (mostly due to the large uncertainty in

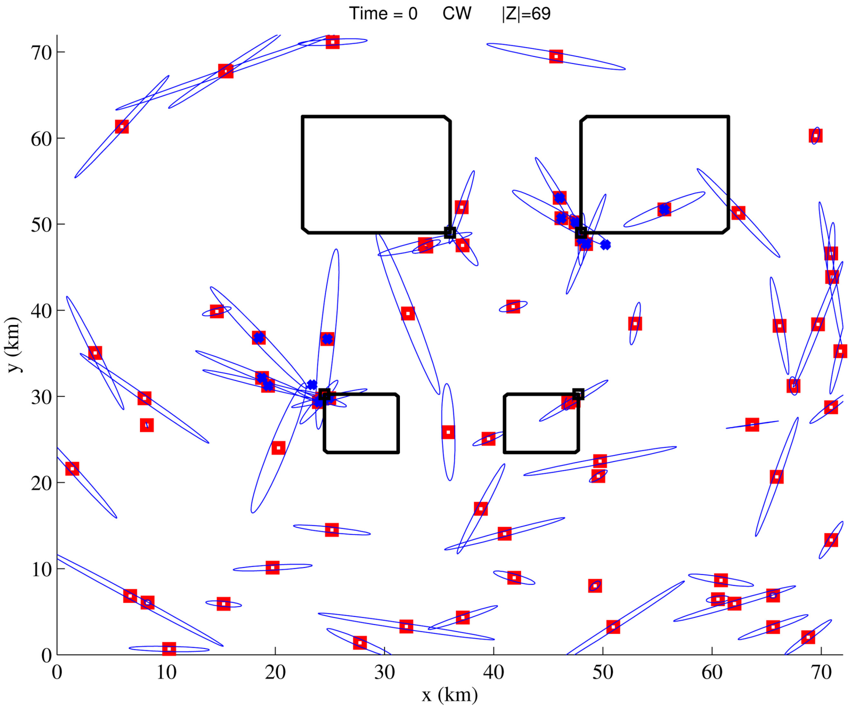


Fig. 3. Winnowed (some of the raw measurements were discarded based on their SNR and Doppler) measurements (red squares) and their covariances (blue ellipses) in the first scan (a blue dot inside a red square denotes a target originated measurement).

the bearing measurements) and can reach 20 km in their major axis. Fig. 3 displays only contacts that survived a detection test on SNR and Doppler, if available.

Predetection fusion is effective at handling both types of difficulties: on one hand, it is designed to preserve target originated measurements while at the same time removing many of the false alarms and on the other hand, its sampling step (described in Section 3) takes advantage of the large measurement uncertainties.

1.3. Improvements for Multitarget Tracking

The goal of multitarget tracking (MTT) is to estimate the states of an unknown and time-varying number of targets from a measurements series produced by a sensor network, despite data association uncertainty, sensor detection uncertainty, false alarms, and noise.

From an implementation perspective, a tracker can be applied in two ways to a multistatic dataset that consists of N sets of contacts from N different sensors at one scan. In the scan-based approach, the N sets of contacts are fed in turn to the tracker. In the fuse-before-track approach [16], all the contacts in a scan are fused and generate a single set of contacts to be fed to the tracker. In the latter approach (the subject of this paper), it is assumed that target originated contacts are more consistent across sensors than noise or clutter returns and thus, tracking performance would be improved.

In the case of the Metron dataset, the first approach, multi sensor scan-based tracking with the GM-CPHD, was not satisfactory [11], while predetection fusion followed by the same GM-CPHD tracker, i.e., the second approach, obtained very good results.

Predetection fusion is so flexible that it can be integrated with a variety of trackers—from well established MTT algorithms such as the Joint Probabilistic Data Association Filter (JPDAF) [1], Multiple Hypothesis Tracking (MHT) [3], and Multiple Frame Assignment (MFA) [24], to approaches recently gaining recognition such as the Maximum Likelihood-Probabilistic Data Association (ML-PDA) [4] tracker and Probability Hypothesis Density (PHD) [20] filter—to drastically reduce the number of input measurements to the tracker and therefore help reduce run time and even obtain improved multisensor tracking performance. For instance, the multisensor CPHD filter [21, 22], which is of $O(nm_1^3 \cdots m_S^3)$ complexity, where n is the number of targets and m_i is the number of measurements from sensor i out of a total of S sensors, would benefit greatly from a reduced number of input measurements.

In the following, we discuss previous attempts at data fusion in sensor networks (Section 2) and provide a detailed description of our algorithm (Section 3). We then present results on a synthetic dataset and on the realistic dataset representative of a large sensor network of low quality sensors (Section 4) and conclude in Section 5.

2. PREVIOUS APPROACHES

In the following, we discuss previous attempts at practical target location-unaware data fusion in sensor networks.

De Theije, et al. [8], [9] presented an algorithm that can be used to fuse two sets of Cartesian contacts observed by two active sonar systems, based on the calculated probability of association between nearest neighbor pairs of contacts, one from each sonar system. The performance of OR and AND fusion rules was evaluated in the presence of position errors in the observations, by means of receiver operating characteristics (ROC) curves. Simulations showed that the benefit of the fusion algorithm was not directly in terms of increased detection performance in the sense of improved ROCs, but instead in enhanced position information of the contacts. The approach is not straightforward to generalize to the case of N sensors. Moreover, it is intuitively easy to see that on datasets with many sensors with low P_{D} and high P_{FA}, such as the Metron dataset, this method would not be able to significantly help a tracker.

Krout and Hanusa [19] analyzed the Metron dataset using their PDA, PDAFAI, and PDAFAIwTS algorithms. In [18], the algorithms were extended to the JPDA and, in order to mitigate the overwhelming amount of false tracks created, a preprocessing step that utilizes a likelihood surface computed over all receivers was introduced prior to tracking. The top 30 local maxima of the final likelihood surface were extracted and sent to the JPDA tracking algorithm as measurements at a particular scan. The results for scenarios 1 and 4 of the Metron dataset showed promise but track fragmentation and track probability of detection were still in need of improvement.

The fuse-before-track architecture (FbT) constitutes an attempt to address the issue of how to best process data in large multisensor surveillance networks with a large number of cheap and limited performance sensors [6]. FbT combines measurement scans through a static fusion operation [17] that leverages more powerful batch processing techniques than can be achieved with scan-based processing. Then, scan-based processing is applied to the output of the static fusion process, enabling real-time surveillance results. Improved performance of FbT processing over centralized tracking has been demonstrated on simulated data [16].

Predetection fusion relies on the FbT approach but improves upon it in a couple of ways. First, the approach described above has been derived only for measurements in Cartesian space while predetection fusion can also easily incorporate Doppler measurements (4D version), and SNR measurements (5D version) which leads to improved accuracy in the fused measurements and helps discriminate between closely spaced targets. Second, measurement covariances are underutilized in the above approach. On the other hand, predetection fusion uses measurement covariances in its initial Monte Carlo sampling step, to alleviate the difficulty introduced by the poor quality of sensors as in the Metron dataset and in EM algorithm step, to further improve accuracy of fused measurements with respect to the true location of the target. Third, our algorithm allows for more sophisticated threshold selection methods.

The multi-hypothesis Generalized Likelihood Ratio Test (GLRT) approach developed by Guerriero, et al. [17] is the natural way to tackle the problem of data fusion in large sensor networks. For each hypothesized target, the location estimate that maximizes the likelihood function is found and the hypothesis with the largest likelihood is selected. Thus, the likelihood function is maximized with respect to both the *number* of targets and their *locations* in Cartesian coordinates.[2]

The drawback of the multi-hypothesis GLRT approach lies in the absence of a penalty mechanism for over-modeling. Therefore, we implement a modified version of this technique, in which the minimum description length (MDL) criterion is used to decide on the number of targets, and compare its performance against predetection fusion. The disadvantages of using the multi-hypothesis GLRT method with MDL penalty are twofold. First, the multi modality of the likelihood surface may induce a loss in performance through missing the global maximum during the optimization step. And second, the computational load is a serious issue.

3. PREDETECTION FUSION WITH POSITION MEASUREMENTS (2D)

In sonar surveillance systems, measurements consist of range, bearing, and possibly Doppler. Range and bearing can be converted into Cartesian measurements. In this version of the algorithm, we consider networks in which Doppler information is not available. As a result, the final fused measurements are two-dimensional (in the xy-plane).

1) *Collection*: All measurements (from all receivers) that arrived at the same time scan are gathered together in one measurement set, on which the following algorithm is run.

2) Sampling: The purpose of this step is to recreate the possible locus of a target, based on the detections hypothesized to have arisen from that target, and use it as motivation for the quantization decisions to be made in the next step of the algorithm.

In large networks of low quality sensors, one expects to encounter considerable measurement errors, as a large bearing error translates into a large and elongated resolution cell at long ranges. For example, in the Metron dataset, the measurements' Cartesian covariance ellipses are very eccentric (see Fig. 3), with some uncertainties as much as 10–20 km (major axis of ellipse).[3]

In order to overcome such large measurements errors, we generate $N_{mc} = 100$ samples via Monte Carlo for each contact, according to the contact's measure-

[2] Details on the multi-hypothesis GLRT can be found in Appendix B.

[3] In this work, the measurement covariances are approximated as elliptical. Given the large bearing error, they are actually banana-shaped. For a better fit, the measurement errors (which are Gaussian in range and in bearing) could be approximated by sums of Gaussians.

ment error covariance matrix. Without this step, a large covariance measurement would still only be seen in the grid cell containing the measurement's nominal value.

Fig. 4 illustrates the need for this step in the Metron dataset. All displayed measurements are target-originated and ideally, all should contribute to the final fused measurement obtained by predetection fusion. Without the sampling step, measurements such as the ones at (24760, 36670) and (18790, 32150) would be quantized to cells far from the true target location that in all likelihood would not pass the detection test described in the thresholding step below. Generation of Monte Carlo samples for these measurements allows all the displayed measurements to be ingested into the EM algorithm (to be described shortly) and thus, to contribute to the final fused measurement.

A similar implementation of this step would be to calculate which cells have edges that intersect the error covariance matrix and count the corresponding contact in those cells. However, a graceful way to find all rectanguloid cells that intersect with a given covariance ellipsoid eludes us; and Monte Carlo sampling is easy.

3) *Sifting*: We then sift these measurement samples according to a grid in the *xy*-plane. When a contact yields at least one sample that is quantized to a grid cell, then that contact is added to the cell's list. Additional MC samples from the same measurement in a given grid cell have no effect.

4) *Thresholding*: A detection is declared in a cell if and only if there are more than τ contacts added to that cell's list. The threshold τ is a tunable parameter and can be computed as follows.

We approximate the sensor probability of false alarms as

$$P_{\mathrm{FA}} = \frac{\text{total number of contacts}}{\text{number of grid cells} \times \text{number of sensors}}. \tag{1}$$

Next, we create the binomial probability mass function that exactly k out of $n = 25$ receivers have detections

$$\Pr(K = k) = \frac{n!}{k!(n-k)!}p^k(1-p)^{n-k} \tag{2}$$

where p is the sensor P_{FA}. More generally, P_{FA} could vary from sensor to sensor. We set the threshold τ by enforcing an upper limit (at designed fused P_{FA}, e.g. 5%) on the false alarm rate of the fused measurements obtained after predetection fusion

$$\min_{\tau}\left\{1 - \sum_{k=0}^{\tau}\Pr(K = k) < \text{designed fused } P_{\mathrm{FA}}\right\}. \tag{3}$$

We test each grid cell's number of hits against the calculated threshold.

5) *Fusion*: For each cell that passes the test, a detection is declared. The cell's listed contacts are then used to refine the estimated measurement location $\hat{x}$ and to estimate the posterior covariance $\hat{R}$. For example, one might use the cell's center for $\hat{x}$ and compute $\hat{R}$ via an assumption of uniformity, but that would give very poor results.

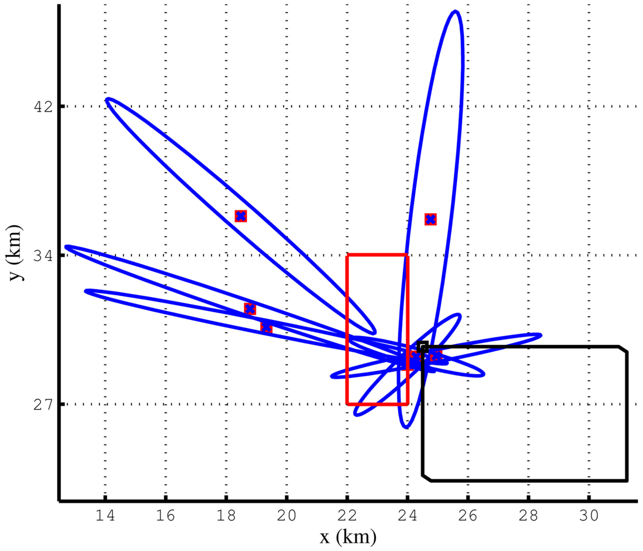


Fig. 4. Sampling step motivating example (Monte Carlo sampling according to each contact's error covariance matrix allows all of these target originated measurements to contribute to the final fused measurement estimate).

Averaging all measurements in a cell's list is a better approach, as demonstrated by the FbT architecture described in Section 2, but it is still far from optimal and remains problematic for the incorporation of Doppler and SNR measurements. More sophisticated approaches, such as the EM algorithm that maximizes $p(X \mid Z)$ over X would seem to be promising alternatives.

The PMHT measurement model is that all the measurements in a cell's list have independent prior probabilities of association that they originated from a target located within that cell or that they are false alarms. Data association à la the PMHT algorithm is a natural choice, as it abandons the generally accepted probabilistic structure of each target having associated at most one measurement at each time. It is a perfectly feasible event that all measurements come from the same target. The PMHT measurement model is a natural fit with EM estimation.

An alternative to this step would be to use the ML estimate of the likelihood calculated based on the measurements in the cell's list as the final fused estimate for the cell, similar to the approach taken by Krout and Hanusa [18], described in Section 2.

In this fusion step, we use the following equations, obtained as per the EM algorithm with a PMHT measurement model described in Appendix A

$$w_i = \frac{\pi_1\mathcal{N}(z_i;\hat{x}_{\mathrm{temp}},R_i)}{\frac{\pi_0}{V} + \pi_1\mathcal{N}(z_i;\hat{x}_{\mathrm{temp}},R_i)} \tag{4}$$

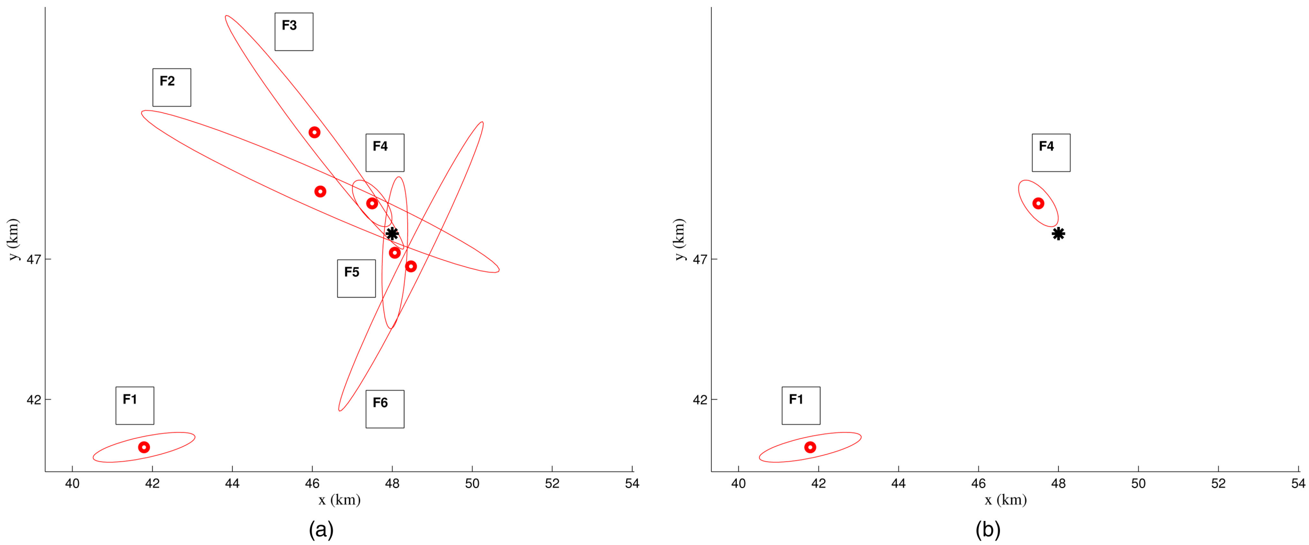


Fig. 5. Merging step notional example. Note that the fused measurement, of small covariance, close to the true location of the target survived merging while redundant fused measurements were removed. (a) Fused measurements before the merging step. (b) Fused measurements that survived the merging step.

$$\hat{x} = \left(\sum_i w_i R_i^{-1}\right)^{-1} \left(\sum_i w_i R_i^{-1} z_i\right) \quad (5)$$

$$\hat{R} = \left(\sum_i w_i R_i^{-1}\right)^{-1} \quad (6)$$

where w_i is the posterior probability that the ith measurement, z_i, comes from the target in that cell, π_1 is the prior probability that the ith measurement comes from the target in that cell, π_0 is the prior probability that the ith measurement comes from clutter (we assume equal priors $\pi_1 = \pi_0 = 0.5$), $\hat{x}_{\text{temp}}$ is the predicted measurement location that is updated in each iteration of the EM algorithm, and V is the volume of a grid cell (in the 2D version, V is the grid cell area in the xy-plane) [31].

We start with $\hat{x}_{\text{temp}}$ at the center of the grid cell that passed the test. In each iteration, we calculate the weights w_i as shown in (4). Then, we can use (5) and (6) to compute $\hat{x}$ and $\hat{R}$ for the declared target. We update $\hat{x}_{\text{temp}}$ with $\hat{x}$ and we repeat for a certain number of iterations.[4]

6) *Merging*: We merge detections that gate with each other, since often neighboring cells have used the same detections from the initial Monte Carlo step. Merging also helps reduce the number of fused measurements that predetection fusion would feed to a tracker while preserving a good target probability of detection.

We test if a fused measurement created in the previous step gates with any of the other fused measurements. Two detections gate with each other if the distance between them (in the xy-plane) is smaller than the diagonal of a grid cell (in the xy-plane). The detection with the smaller fused covariance matrix is kept and the detection with the larger fused covariance matrix is discarded.

Fig. 5 provides an example. In (a), the fused measurements resulting from the fusion step are shown, i.e., the merging step has not yet been applied. In (b), the fused measurements that survived the merging step are shown. Fused measurement F1 survived because it did not gate with any other fused measurement. Fused measurement F2 gated with F3 and since F3 had a smaller covariance, F2 was discarded. F3 gated with F4 and was discarded as F4 had the smallest covariance of all fused measurements. Same for F5 and F6, they were discarded because of gating with F4. Fig. 5 demonstrates how the merging step can be a beneficial addition to predetection fusion: the output fused measurement set is small, yet includes one fused measurement that is very close to the true target location (represented by the black star) and has a small covariance. Such a fused measurement set is desirable as input for any tracker.

3.1. Extensions of Predetection Fusion

Extensions to this work were discussed at length in [14]. Here, we provide an outline for them in order to emphasize that predetection fusion is a flexible, general, and powerful technique.

1) *Incorporating Doppler Measurements*: The 2D version of predetection fusion is to be used when only position measurements are available (no Doppler information is available), e.g., when using a FM waveform. However, multitarget tracking usually operates in a 4-dimensional state space, $[x\ y\ \dot{x}\ \dot{y}]^T$. If Doppler information is available, as in the case of CW waveforms, the 4D version of predetection fusion is a more attractive option because it is able to provide velocity estimates to a tracker, due to its use of Doppler measurements.

[4] Note that there is no constraint that the fused measurement $\hat{x}$ must stay within the grid cell in which it started.

The main differences between the 2D and 4D versions are in the sifting step and the fusion step. As described before, we quantize a sample contact to a grid cell in the xy-plane. Next, we discretize the $\dot{x}\dot{y}$-plane according to a second grid (e.g., 2×2). We calculate possible Doppler values based on the center of the grid cell in the xy-plane the contact was assigned to, all the centers of the grid cells in the $\dot{x}\dot{y}$-plane and the source and receiver location and velocity. We compare the resulting values with the observed Doppler and assign the contact to the cell in the the $\dot{x}\dot{y}$-plane that gave the closest possible Doppler to the observed Doppler.

For the fusion step, the equations used in the 2D version must be modified to incorporate the projection of the range rate (i.e., Doppler information) into velocities in the $\dot{x}\dot{y}$-plane achieved in the modified sifting step just described. For example, (5) would become

$$\hat{x} = \left(\sum_i w_i H_i^T R_i^{-1} H_i\right)^{-1} \left(\sum_i w_i H_i^T R_i^{-1} z_i\right). \qquad (7)$$

Details on the estimation of the measurement matrix, H_i, will be given in another publication [14].

2) *Improvement via SNR Information*: The 4D version of the algorithm relies on Doppler measurements to infer velocity components. The availability of velocity components makes it possible to incorporate aspect information in the fusion step. If SNR measurements are provided in addition to Doppler measurements, the 5D version of predetection fusion should be considered for use.

We quantize a contact to a grid cell the same way as in the 4D version. Then, including SNR information in the predetection fusion algorithm requires a rederivation of the weights of the EM algorithm to have them take into account the likelihood that the contact could have originated from clutter and the likelihood that the contact could have originated from a target. Calculation of these likelihoods entails the evaluation of the predicted contact SNR (based on the SNR model of the dataset) and the observed SNR of the contact.

A related data fusion technique we have subsequently developed, Random Finite Set-based Markov Chain Monte Carlo, has been analyzed and compared against 2D predetection fusion in [15].

4. RESULTS

The predetection fusion algorithm was first tested on a simple synthetic dataset and then on a realistic multistatic sonar dataset of considerably higher difficulty. Additionally, we compared the performance of predetection fusion against that of the (optimal) multi-hypothesis GLRT approach.

4.1. Results on Synthetic Dataset

In this dataset, we assumed 25 identical sensors, with sensor probability of detection $P_D = 50\%$ and nominal standard deviation in two coordinates $\sigma_z = 5000$. The clutter was assumed to be distributed as a homogeneous Poisson process with expected number of false alarms per sensor $\lambda V = 5$.

TABLE I
Simulation Parameters

Number of Sensors	25
Sensor Probability of Detection	0.95
False Alarms per Sensor	2
Delay Error (s)	0.01
Bearing Error (°)	1
Doppler Error (m/s)	0.1
Clutter Doppler Std Dev (m/s)	0.25
Max Doppler (m/s)	6
Number of Cells in x	20
Number of Cells in y	20
Number of Cells in $\dot{x}$	2
Number of Cells in $\dot{y}$	2
Monte Carlo Samples per Contact	100
Designed Fused P_{FA}	0.05
Number of PMHT Iterations	10

Using the same source/receiver grid as in the Metron dataset, we simulated the snapshot in Fig. 6.[5] There are four targets present in the surveillance area, represented in all the following figures by magenta stars. The ellipses represent covariance matrices in Cartesian coordinates; please note their eccentricity and density.

2D predetection fusion was applied. The state space was discretized into 20×20 grid in the xy-plane and the required fused probability of false alarm was 5%.

The results of 2D predetection fusion on the synthetic dataset of Fig. 6 can be seen in Fig. 7, while detailed views of the results can be found in Figs. 8–11. The diameter of the blue dots grows with the number of detections in each cell's list. A circle within the blue dot stands for a declared detection in that particular cell. Red ×s and their corresponding covariance ellipses belong to the contacts generated through predetection fusion.

The 2D version of predetection fusion declared five targets, i.e., it correctly identified the four targets present and it generated one false alarm, which is a satisfactory result given the difficulty of the data. The algorithm estimated target locations accurately, in the vicinity of the true location of the targets and with consistent covariance.

4.2. Performance Comparison

We compared the performance of 2D predetection fusion and the multi-hypothesis GLRT approach using snapshots generated with the parameters in Table I and one target present. The error (i.e., norm) between the predetection fusion estimate and the true target was used as metric of performance.

The static fusion approach [17] that motivated predetection fusion was shown to be dependent on cell size.

[5] In Figs. 6–11, the units of both axes are meters.

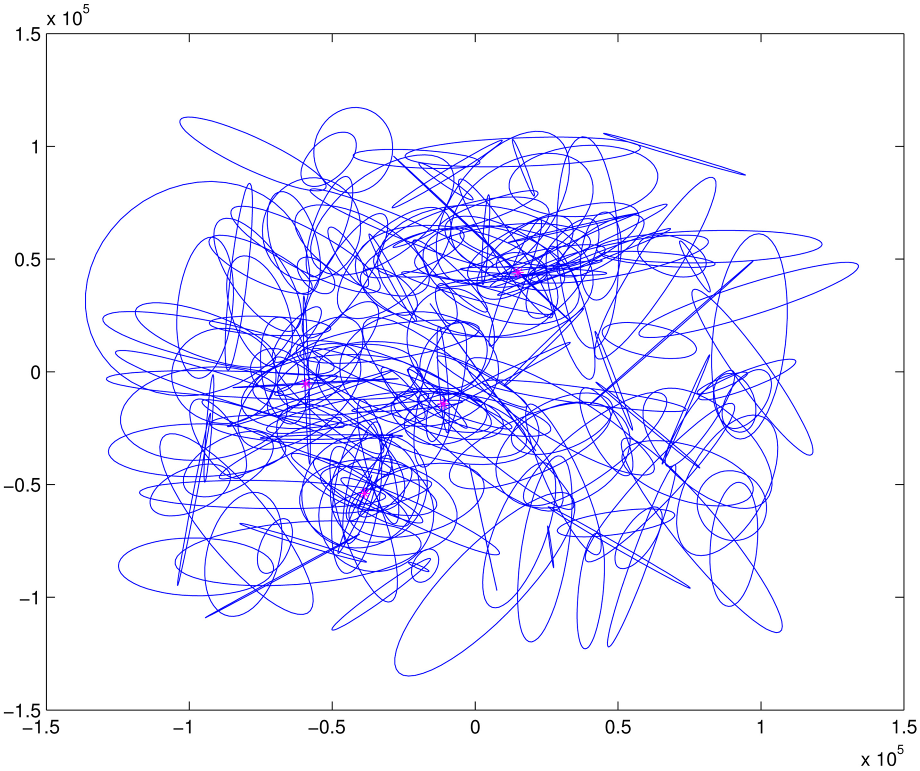


Fig. 6. Setup for synthetic dataset (true target locations in magenta, contacts from all sensors in blue). Note density of contacts and eccentricity of measurement covariances.

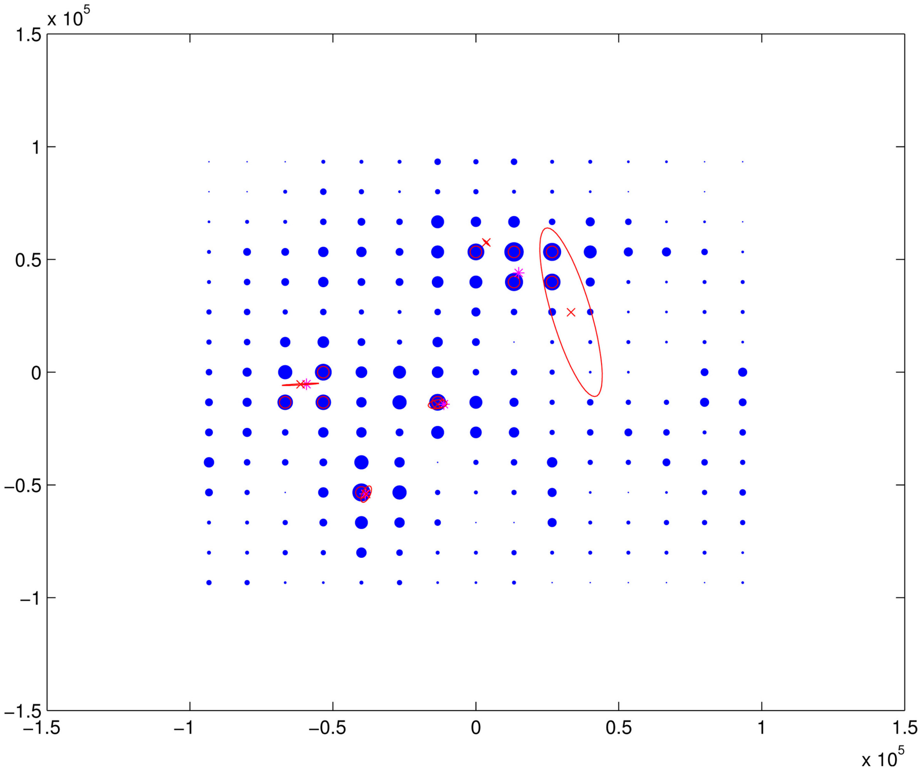


Fig. 7. 2D predetection fusion results in the xy-plane for the synthetic dataset (true target locations in magenta, fused measurements in red, size of blue dots grows with number of detections in cell's list, circle in blue dot stands for declared detection in the cell).

Therefore, we decided to investigate the effect of an increasing number of grid cells in the xy-plane on predetection fusion. We averaged 100 Monte Carlo simulations for each point in Fig. 12 and in each Monte Carlo run, the true position and true velocity of the target were randomly generated. The multi-hypothesis

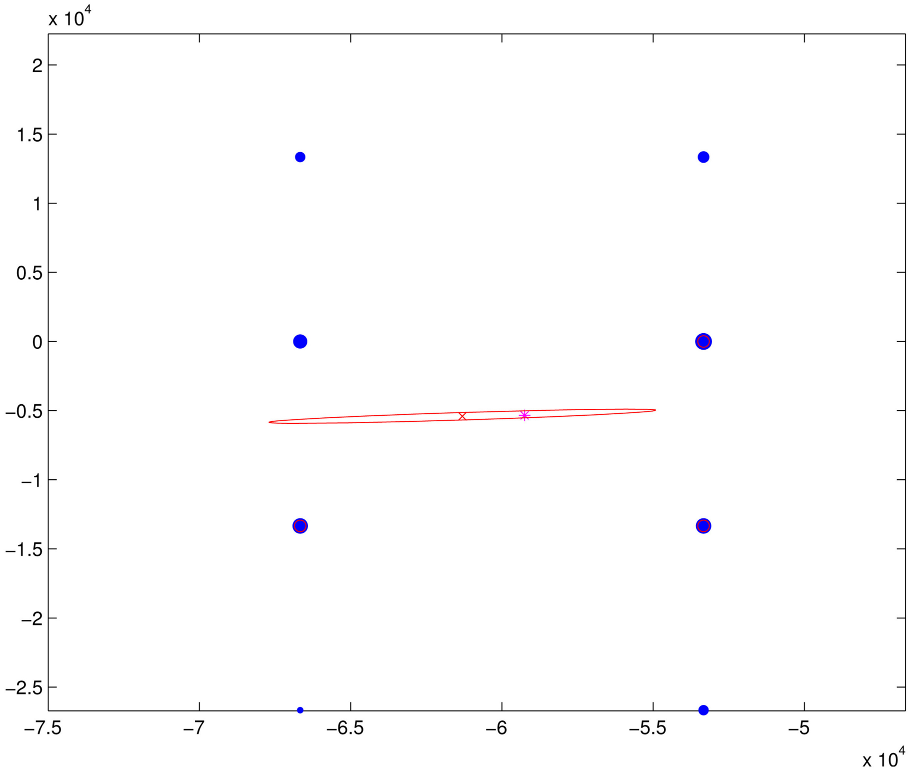


Fig. 8. 2D predetection fusion results in the *xy*-plane for the synthetic dataset: target 1 view.

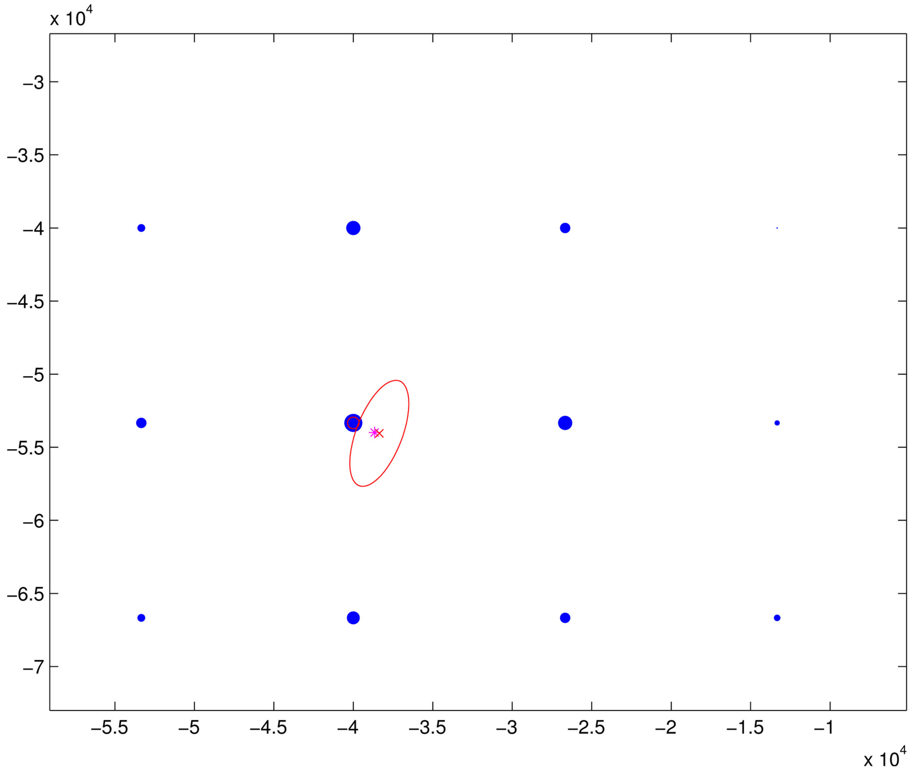


Fig. 9. Results in the *xy*-plane for the synthetic dataset: target 2 view.

GLRT approach is not dependent on the grid cell size.

As expected, the performance of 2D predetection fusion depended strongly on the number of grid cells in the Cartesian plane. By increasing the number of cells in the *xy*-plane, performance significantly improved and seemed to converge. Fusion of the contacts that fell into a smaller cell for which a detection has been declared brings better resolution.

100 Monte Carlo runs were also performed for the multi-hypothesis GLRT approach. It should be noted

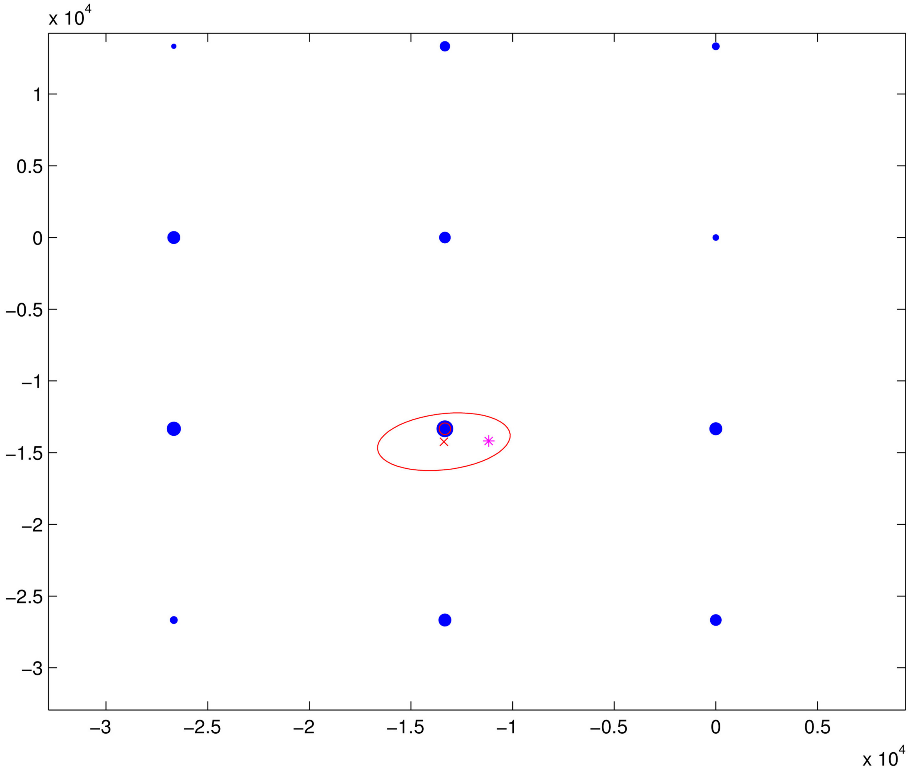


Fig. 10. Results in the *xy*-plane for the synthetic dataset: target 3 view.

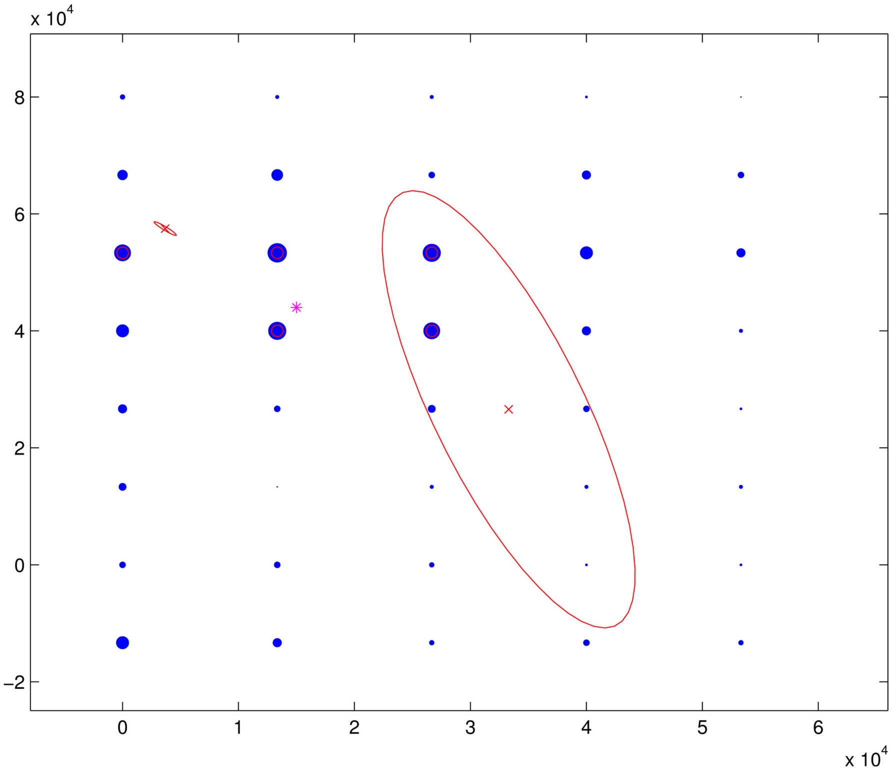


Fig. 11. Results in the *xy*-plane for the synthetic dataset: target 4 view.

that the multi-hypothesis GLRT approach does not depend on grid cell size and therefore its localization error appears as a flat line in Fig. 12. Predictably, the error obtained with the multi-hypothesis GLRT approach is smaller than the error achieved with predetection fusion at all grid cell sizes. The multi-hypothesis GLRT is expected to be optimal (at the expense of a sizeable run time) and it is gratifying to see that predetection fusion

TABLE II
Run Time Versus Clutter Intensity

	$\lambda V = 1$	$\lambda V = 2$	$\lambda V = 3$	$\lambda V = 4$	$\lambda V = 5$
2D	0.054	0.074	0.099	0.112	0.105 sec
GLRT	120.6	281.3	680.7	1052	2125 sec

TABLE III
Run Time Versus Number of Targets

	$N_t = 1$	$N_t = 2$	$N_t = 3$	$N_t = 4$
2D	0.065	0.091	0.128	0.155 sec
GLRT	219.1	317.8	498.8	822.7 sec

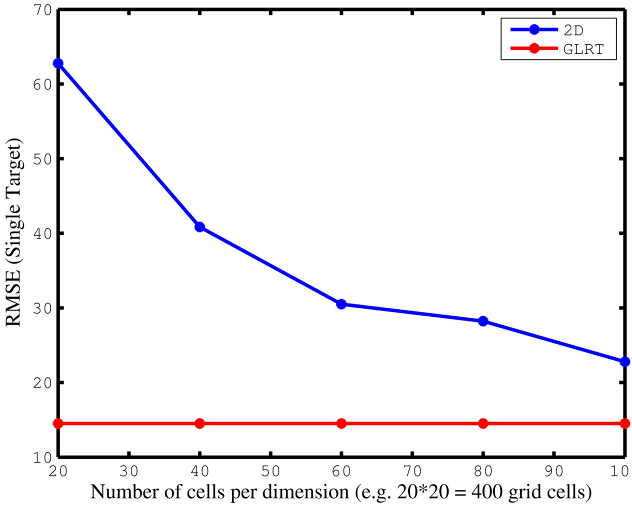


Fig. 12. Effect of number of cells on RMS error for 2D predetection fusion (Multi-hypothesis GLRT, the optimal approach, doesn't depend on cell size).

can approach its performance by using a larger number of grid cells in the Cartesian plane.

Fig. 13 shows that in the 100 Monte Carlo runs, GLRT was always able to detect the single target present at the scene while 2D predetection fusion only missed the target twice when a relatively small number of cells was used to discretize the Cartesian plane.

We also looked at how the run time is affected by increasing clutter density. Table II shows that predetection fusion executes many orders of magnitude faster than the GLRT and the gap grows wider with increasing number of false alarms per sensor per scan, λV. Moreover, Table III shows that predetection fusion executes orders of magnitude faster than the GLRT also for the case of increasing number of targets N_t present in the surveillance area.

On the other hand, the RMS error computed for the target location estimate obtained by predetection fusion that is closest to the true target location is larger than the corresponding RMS error of the GLRT and increases when more targets are added to the scene. A moderately fine grid of 60×60 cells, was used. The GLRT error also becomes larger with an increasing number of targets as seen in Fig. 14.

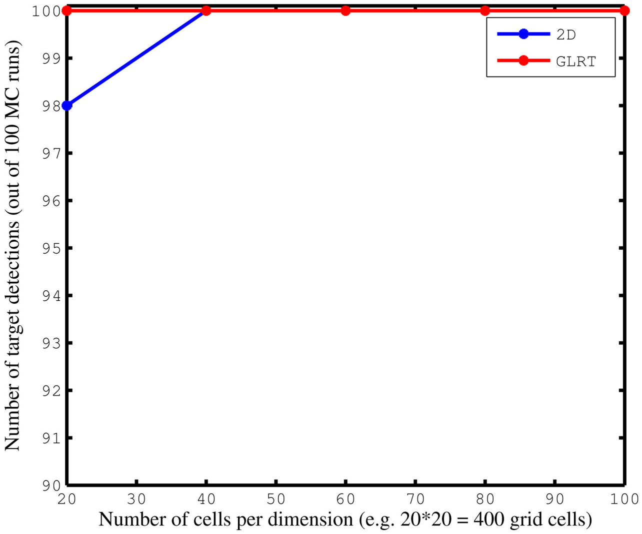


Fig. 13. Number of successful detections of the target (out of 100 MC runs) versus Number of cells in the Cartesian plane (Multi-hypothesis GLRT, the optimal approach, doesn't depend on cell size).

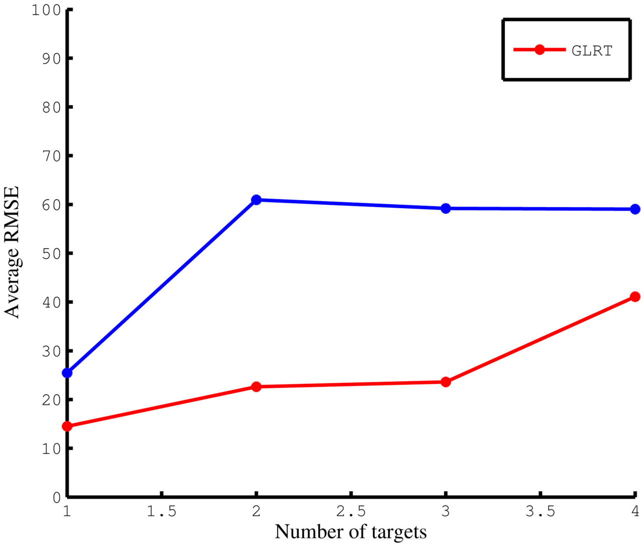


Fig. 14. RMS error for closest estimate to the true location of the target versus Number of targets.

Fig. 15 displays the behavior of the number of times (out of 100 MC runs) predetection fusion correctly estimated the number of targets present with respect to an increasing number of targets. The GLRT, the optimal method, is always able to find the correct number of targets while predetection fusion is close in performance. It should be mentioned that in the few instances in which predetection fusion does not successfully estimate the number of targets, the algorithm underestimates it by one target.

4.3. Results on the Metron Dataset

Fig. 3 showed in red the location in Cartesian space of the measurements in the first scan of data of scenario 1 in the Metron dataset. This set contains measurements

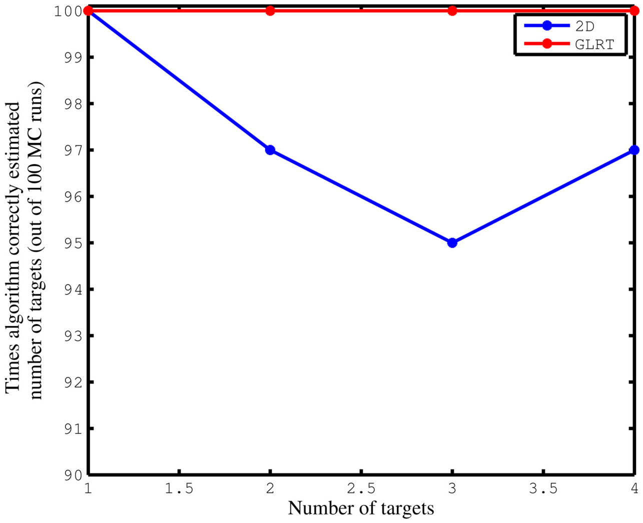


Fig. 15. Number of times (out of 100 MC runs) algorithm correctly estimated the number of targets versus Number of targets.

generated by all the 25 receivers and would serve as input to the predetection fusion algorithm. The corresponding covariances are shown in blue. Note that without the Monte Carlo step inside the predetection fusion algorithm in which 100 samples are generated for each measurement in accordance with its covariance matrix, the tagged measurements (the measurements with a blue dot inside the red square) for target 3 (lower left) would not be associated together.

Fig. 16 shows the fused measurements created through predetection fusion. The 2D version of predetection fusion was applied to the Metron dataset. The number of input contacts to be ingested by the tracker was reduced from 69 (in Fig. 3) to 53 (in Fig. 16). Although there are still many contacts left, note the single low-covariance ones at the southwest corner of target 1, southeast corner of target 2, northeast corner of target 3 and northwest corner of target 4 (the starting positions of the four targets).

In [11], it has been demonstrated that insertion of a predetection fusion step prior to tracking can considerably improve performance. Also in [11], a full set of tracking results obtained with 2D predetection fusion and the GM-CPHD tracker for all five scenarios of the Metron dataset can be found.

5. CONCLUSIONS

Optimal decentralized detection schemes have been researched since the seminal paper of Tenney and Sandell [27]. However, they seem not ready to address many real world problems for radar/sonar applications, in which the location of the target is unknown and does not precisely coincide with a resolution cell grid. Here, we proposed predetection fusion, i.e., a fusion scheme that does not need to know target location a priori.

Predetection fusion was motivated by the need for an efficient way of processing the large amount of data available from sensor networks consisting of many but low performance sensors. We evaluated our algorithm on the challenging Metron multistatic sonar dataset, which is representative of such a configuration. The large number of measurements collected at each scan

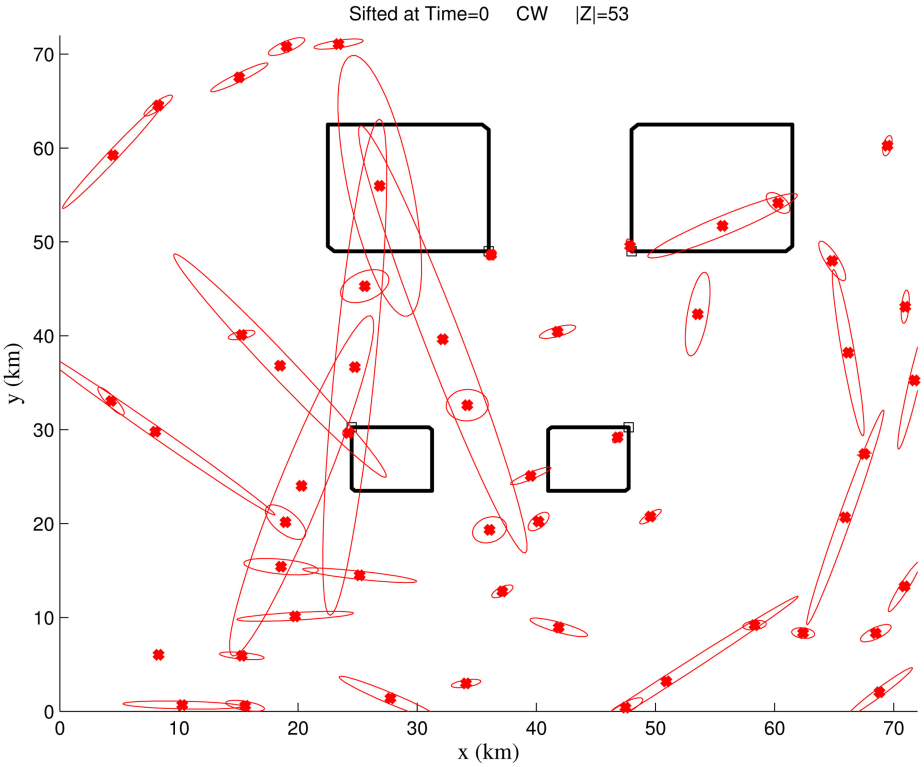


Fig. 16. Fused measurements and their covariances obtained with 2D predetection fusion for the first scan of data.

from the 25 source-receiver pairs (Fig. 2) combined with extremely low sensor quality (P_{D} is about 0.12 per sensor per scan) proved problematic for many well established trackers, such as MHT [7], JPDA [18].

Unsurprisingly, running the GM-CPHD tracker on the Metron data in a centralized approach, i.e., the contacts that arrived at a particular scan from all the 25 sensors were fused into one set of measurements that was sent to the tracker, resulted in considerably long run times and unacceptable performance.

Another way to go about analyzing the Metron dataset would be to apply a multisensor version of the PHD filter. However, the multisensor PHD filter has been derived in detail only for the case of two sensors. Generalization to a large number of sensors is theoretically straightforward but in terms of implementation it would seem that enumerating the partitions of the measurements is intractable when faced with a large sensor network. Running the iterated-corrector approximation, in which the updated multisensor PHD surface is the product of the individual sensor updates, would also be inefficient due to the extremely low sensor probability of detection.

In this paper, we offer an alternative approach, predetection fusion, that while mathematically not as elegant, results suggest that it can circumvent the need for a multisensor PHD. Additionally, predetection fusion is not a tracker dependent technique and can be combined with a variety of trackers to help reduce computation time and obtain improved tracking performance on datasets with a large number of low quality sensors.

Predetection fusion has been developed as a successor to the static fusion approach of Guerriero, et al. [17]. When tested against the optimal multi-hypothesis GLRT approach with MDL penalty, predetection fusion, using a very fine grid for the discretization of the Cartesian plane, performed competitively (Figs. 12 and 14), requiring many orders of magnitude lower run time.

When applied to a difficult, realistic sonar dataset containing a large sensor network composed of sensors of low quality, 2D predetection fusion enabled the GM-CPHD tracker to obtain superior tracking results [11, 18]. In such a setup, the optimal multi-hypothesis GLRT approach with MDL penalty would be impractical.

APPENDIX A. EXPECTATION MAXIMIZATION ALGORITHM

In the E-step of the EM algorithm, the expected value of the complete-data log-likelihood $\log p(X,Y \mid \Theta)$ with respect to unknown data Y given the observed data X and the current parameter estimates Θ^{n-1} is evaluated as

$$Q(\Theta,\Theta^{n-1}) = E[\log p(X,Y \mid \Theta) \mid X,\Theta^{n-1}]. \tag{8}$$

The superscript n indicates the nth EM iteration.

In the M-step, the expectation evaluated in the E-step is maximized

$$\Theta^n = \arg\max_{\Theta} Q(\Theta,\Theta^{n-1}). \tag{9}$$

The two steps are repeated as necessary. More detail on the above equations can be found in [2], which provides a good tutorial on the EM algorithm and its application to parameter estimation for Gaussian Mixture and Hidden Markov Models.

The following description borrows heavily from [31].

Let $X = \{x_s(t)\}$, where $x_s(t)$ is the state of target s at time t, $Z = \{z_r(t)\}$, where $z_r(t)$ is the rth measurement vector at time t, $K = \{k_r(t)\}$, where $k_r(t)$ is the target from which the rth measurement at time t arises. A probabilistic structure for K is needed and it can be assumed that $\Pr(k_r(t) = s) = \pi_s$ and that all are independent random variables. It should be noted that the event in which all measurements come from the same target is perfectly acceptable. Then, (8) can be written as [26]

$$Q(X^{n+1};X^n) = \sum_K \log(p(X^{n+1},K \mid Z))\Pi_{t=1}^T \Pi_{r=1}^{n_t} w_{k_r(t),r}^n(t) \tag{10}$$

$$= \log(\Pi_{s=1}^M p(x_s^{n+1}(1))\Pi_{t=2}^T \Pi_{s=1}^M$$

$$\times p(x_s^{n+1}(t) \mid x_s^{n+1}(t-1)) \tag{11}$$

$$+ \sum_{K,t,r} \log(\pi_{k_r(t)} \mathcal{N}\{z_r(t);\hat{y}_{k_r(t)},R_{k_r(t)}(t)\})w_{k_r(t),r}^n(t) \tag{12}$$

where n_t is the number of measurements at time t, T is the number of time samples in the batch, M is the number of targets, $\mathcal{N}(x;\mu,\Sigma)$ is a Gaussian density in variable x, with mean μ and covariance Σ and $\hat{y}_{k_r(t)}(t) = H_{k_r(t)}(t)x_{k_r(t)}(t)$.

It can be shown that

$$\nabla_{X^{n+1}} Q(X^{n+1};X^n) = \nabla_{X^{n+1}} \hat{Q}(X^{n+1};X^n) \tag{13}$$

where

$$\hat{Q}(X^{n+1};X^n) = \log(\Pi_{s=1}^M p(x_s^{n+1}(1))$$

$$\times \Pi_{t=2}^T \Pi_{s=1}^M p(x_s^{n+1}(t) \mid x_s^{n+1}(t-1)) \tag{14}$$

$$-\frac{1}{2}\sum_{s=1}^M \sum_{t=1}^T [\tilde{z}_s(t) - H_s(t)x_s^{n+1}(t)]^T$$

$$\times \tilde{R}_s(t)^{-1}[\tilde{z}_s(t) - H_s(t)x_s^{n+1}(t)]. \tag{15}$$

Equation (15) is the logarithm of the joint likelihood function of M target models for which there is no data association uncertainty and for which measurements and corresponding measurement covariances are given by their synthetic values $\tilde{z}$ and $\tilde{R}$. In predetection fusion applied to the Metron dataset, we assume one target per grid cell ($s = 1$) and in general a homothetic (or spirograph) PMHT [31], i.e., measurements at scan t can come from any Gaussian density having mean $x(t)$ and variance among $\{R_p\}_{p-1}^{P}$

$$\tilde{z}(t) = \left(\sum_{r=1}^{n_t}\sum_{p=1}^{P} w_{p,r}^{n}(t) R_p^{-1}\right)^{-1} \left(\sum_{r=1}^{n_t}\sum_{p=1}^{P} w_{p,r}^{n}(t) R_p^{-1} z_r(t)\right) \tag{16}$$

$$\tilde{R}(t) = \left(\sum_{r=1}^{n_t}\sum_{p=1}^{P} w_{p,r}^{n}(t) R_p^{-1}\right)^{-1} \tag{17}$$

$$w_{p,r}^{n}(t) = \frac{\pi_p \mathcal{N}\{z_r(t); \hat{y}_p, R_p(t)\}}{\frac{\pi_0}{V} + \sum_{l=1}^{P} \pi_l \mathcal{N}\{z_r(t); \hat{y}_l, R_l(t)\}} \tag{18}$$

meaning that $w_{l,r}^{n}(t)$ is the posterior probability (conditioned on Z and X) the rth measurement at time t is from target $k_r(t)$. In fact, for Metron data, each measurement z_i has its own covariance R_i and hence each measurement can only come from one Gaussian density. Then, $P = 1$ and the above equations simplify to those in Section 3.

APPENDIX B. MULTI-HYPOTHESIS GLRT

The mathematical model for the multi-hypothesis GLRT is described next. $\Lambda_0(Z)$ is the likelihood function given that all measurements are false alarms

$$\Lambda_0(Z) = \prod_{i=1}^{N} \frac{1}{u^{m_i}} \mu_F(m_i) \tag{19}$$

where $Z = \{Z_{ij}\}$ is the measurement set, $i = 1,2,\ldots,N$ is the sensor number, $j = 1,2,\ldots,m_i$ is the measurement number from sensor i, u is the search volume and $\mu_F(.)$ is the probability mass function of the number of false alarms (usually Poisson).

$\Lambda_1(Z \mid \theta_1)$ is the likelihood function given that there is one target

$$\Lambda_1(Z \mid \theta_1) = \prod_{i=1}^{N}\left[\frac{1-P_{\mathrm{D}}}{u^{m_i}}\mu_F(m_i) + \frac{P_{\mathrm{D}}\mu_F(m_i-1)}{u^{m_i-1}m_i}\sum_{j=1}^{m_i} p(Z_{ij} \mid \theta_1)\right] \tag{20}$$

where $p(Z_{ij} \mid \theta_1) = \mathcal{N}(Z_{ij}; \theta_1, \Sigma)$ is the likelihood that measurement j from sensor i originated from target located at θ_1 with Σ as the sensor covariance matrix.

Similarly, $\Lambda_2(Z \mid \theta_1, \theta_2)$ is the likelihood function given that there are two targets

$$\Lambda_2(Z \mid \theta_1, \theta_2) = \prod_{i=1}^{N} p(Z_i \mid \theta_1, \theta_2) \tag{21}$$

where

$$\begin{aligned} p(Z_i \mid \theta_1, \theta_2) = {} & (1-P_{\mathrm{D}})^2 L_i^0 + P_{\mathrm{D}}(1-P_{\mathrm{D}})L_i^1 \\ & + (1-P_{\mathrm{D}})P_{\mathrm{D}}L_i^2 + P_{\mathrm{D}}^2 L_i^{12} \end{aligned} \tag{22}$$

$$L_i^0 = \frac{1}{u^{m_i}}\mu_F(m_i) \tag{23}$$

$$L_i^1 = \frac{\mu_F(m_i-1)}{u^{m_i-1}m_i}\sum_{j=1}^{m_i} p(Z_{ij} \mid \theta_1) \tag{24}$$

$$L_i^2 = \frac{\mu_F(m_i-1)}{u^{m_i-1}m_i}\sum_{j=1}^{m_i} p(Z_{ij} \mid \theta_2) \tag{25}$$

$$\begin{aligned} L_i^{12} = {} & \frac{\mu_F(m_i-2)}{u^{m_i-2}m_i(m_i-1)}\sum_{j=1}^{m_i}\sum_{l=1,l\neq j}^{m_i} p(Z_{ij} \mid \theta_1) \\ & \times p(Z_{il} \mid \theta_2). \end{aligned} \tag{26}$$

Generalization to an arbitrary number of targets is straightforward but tedious to repeat. A constraint is imposed on the maximum number of targets, $T_{\max}$.

With a generalized likelihood ratio test (GLRT) approach, one can find the target location estimates which maximize the likelihood function for each hypothesized number of targets, and choose the largest. The algorithm starts at $\hat{\Lambda}_1(Z) = \max_{\theta_1} \Lambda_1(Z \mid \theta_1)$. The optimal target location estimate is $\hat{\theta}_1 = \arg\max_{\theta_1} \Lambda_1(Z \mid \theta_1)$. The target location estimate $\hat{\theta}_1$ is substituted back into $\hat{\Lambda}_1(Z)$.

If $\hat{\Lambda}_1(Z) < \hat{\Lambda}_0(Z)$, the algorithm stops and declares that no targets are present. Otherwise, the algorithm moves forward to computing $\hat{\Lambda}_2(Z)$ with $\hat{\theta}_1$ fixed as the location estimate for target 1. If $\hat{\Lambda}_2(Z) < \hat{\Lambda}_1(Z)$, the algorithm stops and declares that one target is present. Otherwise, the algorithm continues to $\hat{\Lambda}_3(Z)$ and beyond.

We implemented a modified GLRT, in which the minimum description length (MDL) criterion is used to decide on the number of targets. MDL is described by

$$\hat{t} = \arg\min_{t}\{-\ln \hat{\Lambda}_t(Z) + \tfrac{1}{2} q \ln N\} \tag{27}$$

where $\hat{t}$ is the estimated number of targets, q is the number of independently adjusted parameters in the model (i.e., $q = 2t$ as target location is calculated in two Cartesian coordinates), N is the number of observations and $\ln \hat{\Lambda}_t(Z)$ is the log likelihood achieved at the Maximum Likelihood Estimates (MLEs) of the target locations for the hypothesis that there are t targets present.

REFERENCES

[1] Y. Bar-Shalom and X. R. Li
Multitarget-Multisensor Tracking: Principles and Techniques.
YBS Publishing, 1995.

[2] J. Bilmes
A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian Mixture and Hidden Markov Models.
Technical Report ICSI-TR-97-02, University of Berkeley, 1998.

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

[4] W. Blanding, P. Willett, and Y. Bar-Shalom
ML-PDA: Advances and a new multitarget approach.
EURASIP Journal on Advances in Signal Processing, Article ID 260186, 2008.

[5] Z. Chair and P. K. Varshney
Optimal data fusion in multiple sensor detection systems.
IEEE Transactions on Aerospace and Electronic Systems, **22** (1986), 98–101.

[6] S. Coraluppi and C. Carthel
Multi-stage data fusion and the MSTWG TNO datasets.
In *Proceedings of 12th International Conference on Information Fusion*, Seattle, WA, 2009.

[7] S. Coraluppi and C. Carthel
An ML-MHT approach to tracking dim targets in large sensor networks.
In *Proceedings of 13th International Conference on Information Fusion*, Edinburgh, Scotland, 2010.

[8] P. de Theije and C. van Moll
An algorithm for the fusion of two sets of (sonar) data.
In *Oceans 2005—Europe*, vol. 1, Brest, France, 2005.

[9] P. de Theije, C. van Moll, and M. Ainslie
The dependence of fusion gain on signal-amplitude distributions and position errors.
IEEE Journal Ocean. Eng., **33**, 3 (2008), 266–277.

[10] O. Erdinc
Multistatic Sonar Target Tracking.
Ph.D. thesis, University of Connecticut, 2008.

[11] R. Georgescu, S. Schoenecker, and P. Willett
GM-CPHD and ML-PDA applied to the Metron multi-static sonar dataset.
In *Proceedings of 13th International Conference on Information Fusion*, Edinburgh, Scotland, 2010.

[12] R. Georgescu and P. Willett
The gm-cphd tracker applied to real and realistic multistatic sonar datasets.
IEEE Journal on Oceanic Engineering.

[13] R. Georgescu and P. Willett
The GM-CPHD applied to the corrected TNO-Blind, adjusted SEABAR07 and Metron multi-static sonar datasets.
In *Signal and Data Processing of Small Targets, Proc. SPIE*, vol. 7698, Orlando, FL, 2010.

[14] R. Georgescu and P. Willett
Predetection fusion with doppler measurements and amplitude information.
Accepted to *IEEE Journal of Oceanic Engineering*, 2011.

[15] R. Georgescu and P. Willett
Random finite set markov chain monte carlo predetection fusion applied to multistatic sonar data.
Accepted to *IEEE Transactions on Aerospace and Electronic Systems*, 2011.

[16] M. Guerriero
Statistical Signal Processing in Sensor Networks.
Ph.D. thesis, University of Connecticut, 2009.

[17] M. Guerriero, S. Coraluppi, and P. Willett
Analysis of scan and batch processing approaches to static fusion in sensor networks.
In *Signal and Data Processing of Small Targets, Proc. SPIE*, vol. 6969, Orlando, FL, 2008.

[18] D. Krout and E. Hanusa
Likelihood surface preprocessing with the JPDA algorithm: Metron data set.
In *Proceedings of 13th International Conference on Information Fusion*, Edinburgh, Scotland, 2010.

[19] D. Krout and D. Morrison
PDAFAI vs. PDAFAIwTS: TNO Blind dataset and SEABAR 07.
In *Proceedings of 12th International Conference on Information Fusion*, Seattle, WA, 2009.

[20] R. Mahler
Multitarget Bayes filtering via first-order multitarget moments.
IEEE Transactions on Aerospace and Electronic Systems, **39**, 4 (2003), 1152–1178.

[21] R. Mahler
The multisensor PHD filter: I. General solution via multitarget calculus.
In *Signal Processing, Sensor Fusion and Target Recognition XVIII, Proc. SPIE*, vol. 7336, 2009.

[22] R. Mahler
The multisensor PHD filter: II. Erroneous solution via Poisson magic.
In *Signal Processing, Sensor Fusion and Target Recognition XVIII, Proc. SPIE*, vol. 7336, 2009.

[23] K. Orlov
Description of the Metron simulation dataset for MSTWG. 2009.

[24] A. Poore
Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking.
Computational Optimization and Applications, **3** (1994), 27–57.

[25] A. R. Reibman and L. W. Nolte
Optimal detection and performance of distributed sensor systems.
IEEE Transactions on Aerospace and Electronic Systems, **23** (1987), 24–30.

[26] R. Streit and T. Luginbuhl
Probabilistic multi-hypothesis tracking.
Technical Report 10428, Naval Undersea Warfare Center, 1995.

[27] R. Tenney and N. Sandell
Detection with distributed sensors.
IEEE Transactions on Aerospace and Electronic Systems, **17** (1981), 501–510.

[28] J. N. Tsitsiklis
Decentralized detection.
In *Advances in Statistical Signal Processing, Vol. 2—Signal Detection*, 1993, 297–344.

[29] J. N. Tsitsiklis
Extremal properties of likelihood ratio quantizers.
IEEE Trans. Commun., **41**, 4 (1993), 98–101.

[30] D. Warren and P. Willett
Optimum quantization for detector fusion: Some proofs, examples and pathology.
Journal of the Franklin Institute, **336**, 2 (1999), 323–359.

[31] P. Willett, Y. Ruan, and R. Streit
PMHT: Problems and some solutions.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 4 (2002), 738–754.

Ramona Georgescu received her B.A.s in computer science and physics from Connecticut College in 2004 and her M.Sc. in electrical engineering from Boston University in 2007.

She is currently a Ph.D. candidate in the Electrical and Computer Engineering Department at the University of Connecticut, working under the direction of Dr. Peter Willett. Her area of interest is statistical signal processing, with an emphasis on estimation and multitarget tracking.

Peter Willett (F'03) received his B.A.Sc. (engineering science) from the University of Toronto in 1982, and his Ph.D. degree from Princeton University in 1986.

He has been a faculty member at the University of Connecticut since 1986, and since 1998 has been a professor. He has published 135 journal articles (13 more under review), 290 conference papers, and 9 book chapters. His primary areas of research have been statistical signal processing, detection, machine learning, data fusion and tracking. He has interests in and has published in the areas of change/abnormality detection, optical pattern recognition, communications and industrial/security condition monitoring.

He is editor-in-chief for *IEEE Transactions on Aerospace and Electronic Systems*, and until recently was associate editor for three active journals—*IEEE Transactions on Aerospace and Electronic Systems* (for Data Fusion and Target Tracking) and *IEEE Transactions on Systems, Man, and Cybernetics*, parts A and B. He is also associate editor for the IEEE AES Magazine, editor of the AES Magazine's periodic Tutorial issues, associate editor for ISIF's electronic *Journal of Advances in Information Fusion*, and is a member of the editorial board of IEEE's Signal Processing Magazine. He was a member of the IEEE AESS Board of Governors 2003–2009. He was general cochair (with Stefano Coraluppi) for the 2006 ISIF/IEEE Fusion Conference in Florence, Italy, Program Co-Chair (with Eugene Santos) for the 2003 IEEE Conference on Systems, Man & Cybernetics in Washington, D.C., and program cochair (with Pramod Varshney) for the 1999 Fusion Conference in Sunnyvale. He was coorganizer of the tracking subsession at the 1999 IEEE Aerospace Conference, and has been organizer of the Remote Sensing Track of that conference 2000–2003. Jointly with T. Kirubarajan he has coorganized the SPIE "System Diagnosis and Prognosis: Security and Condition Monitoring Issues" Conference in Orlando, 2001–2003. He has been a member of the IEEE Signal Processing Society's Sensor-Array & Multichannel (SAM) Technical Committee since 1997, and both serves on that TC's SAM Conferences' Program Committees and maintains the SAM website.

Stefano Marano received the Laurea degree in electronic engineering (cum laude) and the Ph.D. degree in electronic engineering and computer science both from the University of Naples, Italy, in 1993 and 1997, respectively.

Currently, he is a professor at the University of Salerno, Italy, where he was formerly assistant professor. His areas of interest include statistical signal processing with emphasis on inference, sensor networks, and information theory. He has coauthored more than 70 papers about these and related topics, including some invited, mainly on international journals/transactions and proceedings of international conferences. Professor Marano was corecipient of the S. A. Schelkunoff Transactions Prize Paper Award for the best paper published in the *IEEE Transactions on Antennas and Propagation* in 1999. Recently, he is/was in the Scientific Committee of the Remote Sensing Laboratory for Environmental Hazard Monitoring (ReSLEHM), University of Salerno, and in the Organizing Committee of the Ninth International Conference on Information Fusion (FUSION 2006).

Vincenzo Matta received the Laurea degree in electronic engineering and the Ph.D. degree in information engineering from University of Salerno, Fisciano, Italy, in 2001 and 2005, respectively.

He is currently an assistant professor with the University of Salerno. His main research interests include detection and estimation theory, signal processing, wireless communications, multiterminal inference and sensor networks.

A Probabilistic Computational Model for Identifying Organizational Structures from Uncertain Activity Data

XU HAN
FEILI YU
GEORGIY LEVCHUCK
KRISHNA PATTIPATI
FANG TU

The knowledge of the principles and goals under which an adversary organization operates is required to predict its future activities. To implement successful counter-actions, additional knowledge of the specifics of the organizational structures, such as command, communication, control, and information access networks, as well as responsibility distribution among members of the organization, is required. Our focus here is on identifying the mapping between hypothesized nodes of an adversary command organization ("model network") and tracked individuals, resources and activities ("data network"). We formulate the organizational structure identification problem as one of associating the nodes of the noisy data network with the nodes of the model network. The problem of minimizing the negative log likelihood ratio with respect to the mapping versus null mapping (thereby capturing the possibility that no hypothesized model network is a good match) leads to a Quadratic Assignment Problem (QAP). We solve the QAP using what we call an iterative *m*-best soft assignment algorithm, combining Bertsekas' auction algorithm and Murty's *m*-best assignment algorithms in a novel way. The experimental results show that our probabilistic model and the *m*-best soft assignment-based algorithm can accurately identify the different organizational structures and achieve correct node mappings among organizational members under uncertainty. We also apply the *m*-best soft assignment algorithm to the general QAP and compare its performances to the hitherto best solutions.

Manuscript received June 14, 2010; revised March 17, 2011 and January 26, 2012; released for publication January 28, 2012.

Refereeing of this contribution was handled by Mujdat Cetin.

This work was supported by the U.S. Office of Naval Research under Grant N00014-09-1-0062 and by Aptima under subcontract 0565-1494.

Authors' addresses: X. Han and K. Pattipat, University of Connecticut, Department of Electrical and Computer Engineering, Storrs, CT 06269; F. Yu, Science Application International Corporation, Madison, WI 53719; G. Levchuck, Aptima Inc., Woburn, MA 01801; F. Tu, GE Healthcare, WI 53188.

1557-6418/12/$17.00 © 2012 JAIF

1. INTRODUCTION

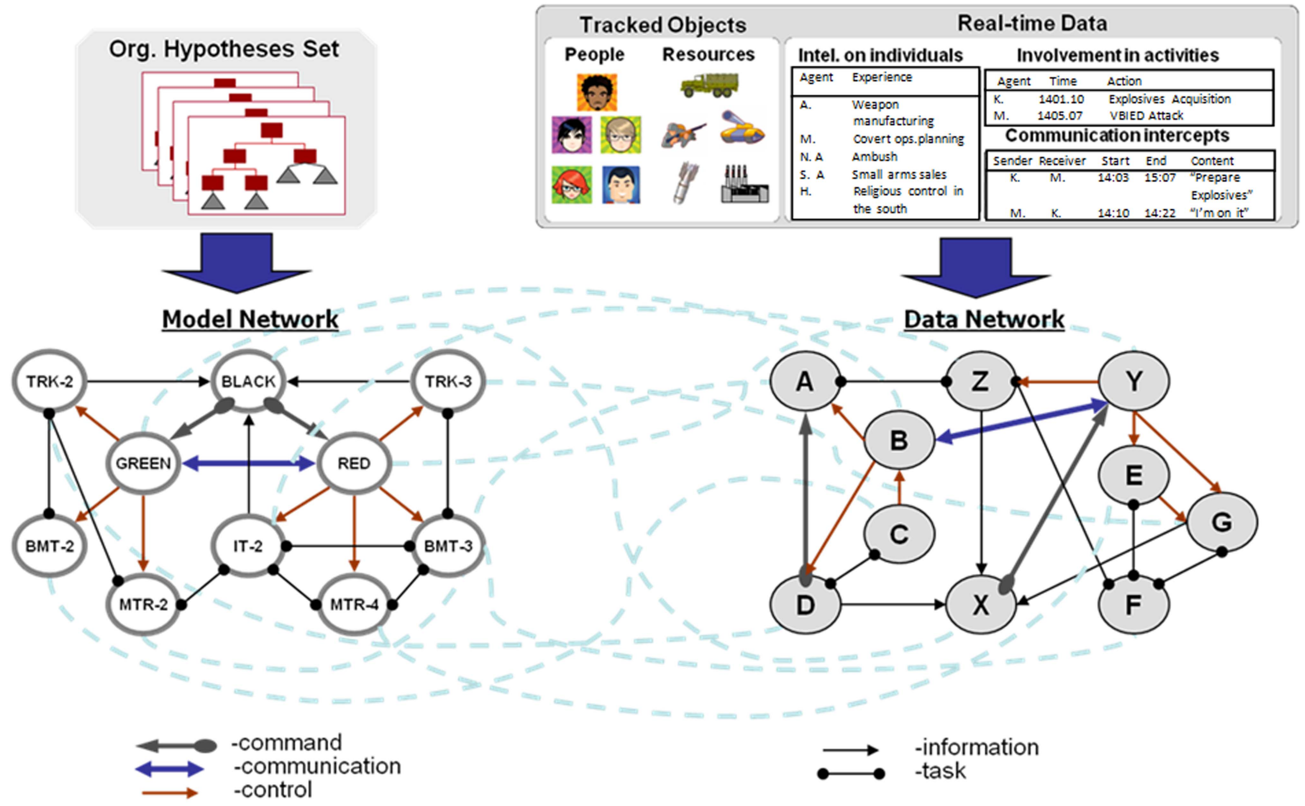
1.1. Motivation

Analysis of the structures of organizations, ranging from the more structured command systems of a conventional military to the decentralized and elusive adversary organizations, such as insurgent and terrorist groups, suggests that strong relationships exist between the structure, resources, and objectives of those organizations and the resulting actions [31]. The organizational members act in their assigned missions by accomplishing tasks and these activities may leave detectable clues or events in the information space. The dynamic evolution of these events creates patterns of organizational activities, which may be related, linked, and tracked over time [39], [46]. More significantly, the patterns can be used to discover the underlying organizational structure. We mean by organization discovery the ability to recognize the command, control, communication, and task structures of the organization. However, the challenge is that most of the time we cannot observe the elements of the structures of the organization. Instead, we can obtain uncertain transaction data involving the activities of organizational members. The specific activities depend on the structure of enemy command and control (C^2) organization which, in turn, depends on the goals of that organization.

As an illustrative example, consider the organizational structure identification problem shown in Fig. 1. The hypothesized model network represents an adversary organization whose members are comprised of bomb makers (BMT), mortars (MTR), intelligence teams (IT), truck drivers (TRK), and commanders (black, red, and green). To identify this network, all the collected observations are linked together to form the data (observed) network shown in Fig. 1. We need to map the nodes of this data network to the model network—a hypothesized C^2 organization with a specified command, communication, control and task substructures. In Fig. 1, 10 nodes of the adversary network have been detected (A, B, C, D, E, F, G, X, Y, Z), and the concomitant communication intercepts and observed activities of the adversary are aggregated into the data network shown. Matching the nodes of this network to the nodes of a hypothesized C^2 model network produces the following association: A = MTR-2, B = GREEN, C = BMT-2, D = TRK-2, E = MTR-4, F = BMT-3, G = TRK-3, X = BLACK, Y = RED, Z = IT-2. That is, we say that tracked agent "X" is commander "BLACK," tracked resource "A" is a mortar resource (MTR-2), agent "Y" is commander "RED," and so on. This paper provides an analytic framework for addressing this network matching problem.

1.2. Literature Review

The nodes in a model network represent the entities of interest (humans, agents, assets, place, etc.) and



Intel. on individuals

Agent	Experience
A.	Weapon manufacturing
M.	Covert ops.planning
N. A	Ambush
S. A	Small arms sales
H.	Religious control in the south

Involvement in activities

Agent	Time	Action
K.	1401.10	Explosives Acquisition
M.	1405.07	VBIED Attack

Communication intercepts

Sender	Receiver	Start	End	Content
K.	M.	14:03	15:07	"Prepare Explosives"
M.	K.	14:10	14:22	"I'm on it"

Fig. 1. Illustrative example of identifying an organizational structure.

edges denote relationships or connections among them (interactions, communications, influences, geographical adjacencies, etc.). Current approaches to detect and analyze network/graph structures have their roots in social network analysis (SNA), link discovery (LD) and relational data mining (RDM). SNA explores the structures of groups in a society by modeling individuals, places, and objects as nodes of a graph, and adding links between nodes to represent relations among them [52], [45]. SNA can also be employed to identify the key nodes in a network and has applications in crime analysis as well as professional and social network services, such as LinkedIn and Facebook [18], [38]. Link discovery has its roots in data mining; it is concerned with the discovery of relational patterns that indicate potentially interesting activities based on a large amount of relational data [37]. The focus of RDM is on extracting connections in data based on multiple relational tables that are richly connected [37], [20]. Evidently, link discovery employs relational data mining.

The current network identification problems can be categorized into two classes: supervised and unsupervised. By "*supervised*," we mean network identification problems with a prior knowledge of *possible output (model) networks* as opposed to "*unsupervised*" ones where there is no such knowledge of model networks. This paper employs supervised network matching in that we assume the existence of a library of hypothesized model networks and seek to obtain the most likely network in the library based on the observed data network. The unsupervised network identification problem, on the other hand, learns the network structure and dynamics based on a number of observed data networks. Thus, in the unsupervised case, the focus is on characterizing the statistical properties of observed data networks.

1) *Unsupervised network identification*: Graph representations are widely used for dealing with structural information in different domains, such as social networks, probabilistic sampling designs [16], image processing, and pattern recognition. In the context of SNA, one important application is to identify the key members in an organization by computing the so-called centrality measures [17]. A directed graph (digraph) model is employed to study the centrality measures, such as the *degree*, *betweenness* and *closenesss* [52]. Probabilistic graph models, such as Holland-Leinhardt model, the $p*$ model, and Markov random graphs [16], are used to infer whether there exists a link (edge) from a node i to another node j. Holland-Leinhardt model is an *a posteriori* blocking procedure in the framework of the exponential family [28]. The $p*$ model is a simplified Markov random graph with binary attributes [49]. In this model, the network identification problem reduces to one of estimating the adjacency matrix associated with the digraph via a maximum likelihood technique [11].

Frank considered a more general graph model, termed *valued digraph* model, to handle attributed graphs [16]. Coffman and Marcus combined digraph and Hidden Markov Models (HMM) to track the dynamic evolution of groups [9]. Inductive Logic Programming (ILP) was recently used by Mooney, et al., [37] for link discovery and relational data mining. Here, the relational database tables are translated into first-order logic and inference based on the rules of this logic is performed, given a database of background facts and logical definitions of relations. ILP does not involve any probabilistic relational concepts; therefore, it does not capture the uncertain nature of organizational structures and processes. Furthermore, it suffers from large computational demands stemming from the need to search for a solution in a large space of structural and relational hypotheses.

2) *Supervised network identification*: Given a library of hypothesized networks, supervised network identification is a more realistic alternative to unsupervised network identification. A general graph model, termed an Attributed Relational Graph (ARG), composed of multi-attributed nodes and multi-attributed links (edges), is widely used in pattern recognition and graph matching. In order to achieve good correspondence (association, matching) between two attributed relational graphs, measures that adequately represent the similarity between the attributes of nodes and the similarity between attributes of edges should be defined. Many of the early efforts on graph matching define the edit distance between two graphs, viz., the number of modifications that one needs to make to change one graph to the other, as a similarity measure; the smaller the distance the greater is the similarity [13], [14], [48].

In recent years, two categories of modeling efforts are attracting increasing attention. The first is based on deterministic linear least squares and graph eigen space projection, which is also termed the spectral graph theory [43], [7], [44]. This is a family of techniques that aim to characterize the global structural properties of graphs using the eigenvalues and eigenvectors of the adjacency matrix [7]. The eigenvalues of a graph are intimately connected to important structural features, and the associated eigenvectors can be used to discover the clusters and other local features, such as node and link attributes. Scott [43] showed how to recover correspondences via singular value decomposition (SVD) on the point association matrix between different images. Shapiro [44] extended this method to a point proximity matrix, which is constructed by computing the Gaussian weighted distance between points. However, these methods have their limitations: they require the two graphs to be of equal size. In addition, when the two graphs are large, the eigenvalue methods are computationally expensive.

The second class of methods employs a probabilistic approach, such as probabilistic relaxation labeling, and Markov random fields. The probabilistic methods assume that the structure is defined probabilistically for graph elements and their relations. The identification of structure involves optimizing a likelihood function that quantifies the match between a hypothesized graph and the observations. Using a probabilistic relaxation framework, Chrismas, et al. [5] have developed a statistical model for pair-wise attribute relations. Hancock and Kittler [26], [29] use an iterative approach, called probabilistic relaxation, to take into account binary relations. Wilson [50] used a Bayesian framework to determine the compatibility coefficients required for performing graph matching by probabilistic relaxation. In [15], the objective function is a series of exponential functions of the Hamming distances between graph neighborhoods, and in [34], a super clique is defined as a clique containing a node and its neighboring nodes. These efforts led to the application of Markov Random Field (MRF) theory to graph matching problems. Other related works are [54] and [4].

1.3. The Organization and Scope of Paper

In this paper, we use attributed relational graphs (ARGs) for representing the model and data networks. We employ negative log likelihood ratio of the mapping versus null mapping (all the observations are false alarms in the sense that the data does not originate from any hypothesized model network) to derive an energy function that serves as a scoring function. The resulting problem corresponds to a quadratic assignment problem (QAP). We solve the NP-hard QAP via a series of m-best linear assignment problems, termed the m-best soft assignment algorithm (m-Best SAA). We demonstrate that our m-Best SAA has the capability to discover hidden organizational structures from real data sets, and that it can be used to solve general QAPs as well.

The paper is organized as follows. In Section 2, we formulate the organizational identification problem as a standard QAP. In Section 3, a review of QAP algorithms is given and m-Best SAA is proposed. In Section 4, we provide computational results for the network identification problem and for general QAPs. Finally, the paper concludes with summary and open topics for future research.

2. MODELING AND PROBLEM FORMULATION

The problem of structural discovery in practice is very complex: the observed data do not relate to the structure directly; instead, they relate to their manifestation in the form of activities and processes that are enabled by the organizational structure(s) and performed by the organization's members. Therefore, the algorithms to reconstruct the organization from observations alone would need to search through a very large space of possible structures. Given historical data and the availability of subject-matter experts, we can instead pose the problem as one of hypothesis testing. Here, a set of predefined hypotheses about the adversary organization and its sub-elements (model networks) is given. The problem then becomes one of rank ordering these

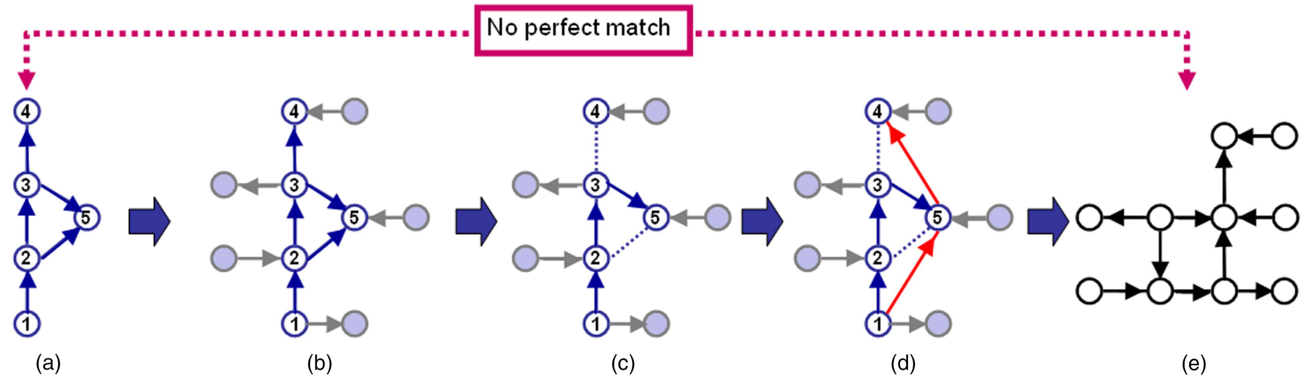


Fig. 2. Process of generating data from model network: (a) model network (b) irrelevant observations (c) missing data (d) errors/deceptions (e) data network.

predefined hypotheses on the basis of how best they match (or explain) the observed data network. We employ likelihood ratio between a given mapping and null mapping (the hypothesis that no model network is a good match for the observed data network) as the network matching criterion. For H hypotheses, one needs to solve H network matching problems. The hypothesis with the best score is the most likely organization that could have generated the observed data network.

2.1. Likelihood Ratio Criterion

Consider two attributed graphs, $G_M = (V_M, E_M)$ and $G_D = (V_D, E_D)$, representing a hypothesis graph (or model network) and a data (observation) network, respectively. The sets $\{V_M, E_M\}$ and $\{V_D, E_D\}$, represent the set of vertices (nodes) and edges (links) of the model network and data network, respectively. A single node in network G_M is denoted by p or q and that in network G_D by i or j.

Each node in G_M is specified by an attribute vector:

$$\underline{a}_M^{(p)} = [a_p^{(1)}, \dots, a_p^{(K_1)}]^T; \qquad 1 \le p \le |V_M|. \tag{1}$$

Thus, there are K_1 attribute types associated with our graph matching problem. As the example in Fig. 1, the relations derived from pairs of vertices in G_M can be undirected (e.g. geographical adjacencies, task etc.) or directed (e.g. command and control, communication, information etc.). We replace each undirected link by two directed links, and thus define both G_M and G_D as directed graphs. The relations derived from pairs of vertices $\langle p,q \rangle$ are:

$$\underline{a}_M^{\langle p,q \rangle} = [a_{p,q}^{(1)}, \dots, a_{p,q}^{(K_2)}]^T; \qquad 1 \le p,q \le |V_M|. \tag{2}$$

Thus, there are K_2 different types of relations between each pair of nodes $\langle p,q \rangle$. We call q an *out-neighbor* (*successor*) of node p [7]. Let $N^+(p)$ denotes the set of *out-neighbors* (*out-list*) of node p.

In the same vein, a node i in the data network has K_1 attributes

$$\underline{a}_D^{(i)} = [a_i^{(1)}, \dots, a_i^{(K_1)}]^T; \qquad 1 \le i \le |V_D| \tag{3}$$

and the pairs of nodes $\langle i,j \rangle$ in the data network have K_2 relations

$$\underline{a}_D^{\langle i,j \rangle} = [a_{i,j}^{(1)}, \dots, a_{i,j}^{(K_2)}]^T; \qquad 1 \le i, \quad j \le |V_D|. \tag{4}$$

We denote the attributes of model and data networks as:

$$\begin{aligned} A_M &= \{\{\underline{a}_M^{(p)}\}, \{\underline{a}_M^{\langle p,q \rangle}\}\}; \qquad \forall 1 \le p, \quad q \le |V_M| \\ A_D &= \{\{\underline{a}_D^{(i)}\}, \{\underline{a}_D^{\langle i,j \rangle}\}\}; \qquad \forall 1 \le i, \quad j \le |V_D|. \end{aligned} \tag{5}$$

We define an assignment matrix $X = [x_{ip}]$, $1 \le i \le |V_D|$, $1 \le p \le |V_M|$, such that $x_{ip} = 1$ implies node i in V_D is mapped to node p in V_M. Evidently, if $x_{ip} = 1$ *and* $x_{jq} = 1$, then edge $\langle i,j \rangle$ in E_D is mapped to edge $\langle p,q \rangle$ in E_M and edge $\langle j,i \rangle$ in E_D is mapped to edge $\langle q,p \rangle$ in E_M.

We assume that: (i) the observations of both node and edge attributes are corrupted by multivariate Gaussian noise; (ii) For different nodes and edges, the noise processes are conditionally independent. Thus, we have

$$\begin{aligned} \underline{a}_D^{(i)} &= \underline{a}_M^{(p)} + \underline{e}_1 \\ \underline{a}_D^{\langle i,j \rangle} &= \underline{a}_M^{\langle p,q \rangle} + \underline{e}_2 \end{aligned} \tag{6}$$

where $\underline{e}_1 \sim N(\underline{0}, \Sigma_1)$ and $\underline{e}_2 \sim N(\underline{0}, \Sigma_2)$ are Gaussian noise vectors with zero mean and covariance matrices Σ_1 and Σ_2, respectively. The likelihood potentials are

$$\begin{aligned} -\log p(\underline{a}_D^{(i)} \mid \underline{a}_M^{(p)}) &= (\underline{a}_D^{(i)} - \underline{a}_M^{(p)})^T (2\Sigma_1)^{-1} (\underline{a}_D^{(i)} - \underline{a}_M^{(p)}) \\ -\log p(\underline{a}_D^{\langle i,j \rangle} \mid \underline{a}_M^{\langle p,q \rangle}) & \\ = (\underline{a}_D^{\langle i,j \rangle} - \underline{a}_M^{\langle p,q \rangle})^T & (2\Sigma_2)^{-1} (\underline{a}_D^{\langle i,j \rangle} - \underline{a}_M^{\langle p,q \rangle}). \end{aligned} \tag{7}$$

The generation of data network is illustrated in Fig. 2. If the data originate from a model network, then the data network can be interpreted as a collection of noisy observations on the model network, i.e., the model network attributes plus observation noise, including false alarms as in Fig. 2(b), missed detections as in Fig. 2(c), and deceptions as in Fig. 2(d). If all the nodes and links in the data network are spurious in the sense that they do not originate from any hypothesized network, we associate the data network to a *null mapping*, denoted by ϕ.

2.2. Likelihood Ratio Criterion

Now, we express the objective of network matching problem as one of maximizing the likelihood ratio of a match as specified by the assignment matrix $X = [x_{ip}]$, to null mapping ϕ, as in [41] and [1]. Assuming that the spurious nodes and spurious edges are uniformly distributed with volume parameters Φ_1 and Φ_2, and P_d $(0 < P_d < 1)$ denoting the detection probability of a node or an edge (Fig. 2(c)),[*]the problem can be written as:

$$
\begin{aligned}
X^* &= \arg\max_X \frac{p(A_D \mid X, A_M)}{p(A_D \mid \phi)} \\
p(A_D \mid X, A_M) &= \prod_{p=1}^{|V_M|}\left\{\left(\prod_{i=1}^{|V_D|}(P_d p(\underline{a}_D^{(i)} \mid \underline{a}_M^{(p)}))^{x_{ip}}\right)(1-P_d)^{(1-\Sigma_{i=1}^{|V_D|}x_{ip})}\right\}\prod_{i=1}^{|V_D|}\left\{\left(\frac{1}{\Phi_1}\right)^{(1-\Sigma_{p=1}^{|V_M|}x_{ip})}\right\} \\
&\quad \cdot \prod_{p=1}^{|V_M|}\prod_{q\in N^+(p)}\left\{\left(\prod_{i=1}^{|V_D|}\prod_{j\in N^+(i)}(P_d p(\underline{a}_D^{\langle i,j\rangle} \mid \underline{a}_M^{\langle p,q\rangle}))^{x_{ip}x_{jq}}\right)(1-P_d)^{(1-\Sigma_{i=1}^{|V_D|}\Sigma_{j\in N^+(i)}x_{ip}x_{jq})}\right\} \\
&\quad \cdot \prod_{i=1}^{|V_D|}\prod_{j\in N^+(i)}\left\{\left(\frac{1}{\Phi_2}\right)^{(1-\Sigma_{p=1}^{|V_M|}\Sigma_{q\in N^+(p)}x_{ip}x_{jq})}\right\} \\
p(A_D \mid \phi) &= \left(\frac{1}{\Phi_1}\right)^{|V_D|}\left(\frac{1}{\Phi_2}\right)^{|E_D|} \\
\Rightarrow X^* &= \arg\max_X \frac{\prod_{p=1}^{|V_M|}\left\{\left(\prod_{i=1}^{|V_D|}(P_d p(\underline{a}_D^{(i)} \mid \underline{a}_M^{(p)}))^{x_{ip}}\right)(1-P_d)^{(1-\Sigma_{i=1}^{|V_D|}x_{ip})}\right\}\prod_{i=1}^{|V_D|}\left\{\left(\frac{1}{\Phi_1}\right)^{(1-\Sigma_{p=1}^{|V_M|}x_{ip})}\right\}}{\left(\frac{1}{\Phi_1}\right)^{|V_D|}} \\
&\quad \cdot \frac{\prod_{p=1}^{|V_M|}\prod_{q\in N^+(p)}\left\{\left(\prod_{i=1}^{|V_D|}\prod_{j\in N^+(i)}(P_d p(\underline{a}_D^{\langle i,j\rangle} \mid \underline{a}_M^{\langle p,q\rangle}))^{x_{ip}x_{jq}}\right)(1-P_d)^{(1-\Sigma_{i=1}^{|V_D|}\Sigma_{j\in N^+(i)}x_{ip}x_{jq})}\right\} \cdot \prod_{i=1}^{|V_D|}\prod_{j\in N^+(i)}\left\{\left(\frac{1}{\Phi_2}\right)^{(1-\Sigma_{p=1}^{|V_M|}\Sigma_{q\in N^+(p)}x_{ip}x_{jq})}\right\}}{\left(\frac{1}{\Phi_2}\right)^{|E_D|}}
\end{aligned}
\tag{8}
$$

where X^* is the optimal assignment matrix. If $|V_M| < |V_D|$, we augment the data network with $|V_M|$ null nodes and assign each node $(1 \le p \le |V_M|)$ in the model network to a distinct node in the data network. The assignments to null nodes in the data network correspond to invalid mappings and indicate missed detections while the unassigned nodes in the data network represent false alarm nodes. Therefore, the constraints are written as:

$$
\begin{aligned}
&\sum_{i=1}^{|V_D|+|V_M|} x_{ip} = 1 \qquad \forall p = 1,\ldots,|V_M| \\
&\sum_{p=1}^{|V_M|} x_{ip} \le 1 \qquad \forall i = 1,\ldots,|V_M| + |V_D|; \quad x_{ip} \in \{0,1\}.
\end{aligned}
\tag{9}
$$

On the other hand, if $|V_M| \ge |V_D|$, we add $|V_D|$ null nodes to the model network and associate each node in the data network with a distinct model network node. The unassigned model network nodes represent missed detections and the model network nodes mapped to null nodes imply spurious measurements. In this case, we have constraints as:

$$
\begin{aligned}
&\sum_{p=1}^{|V_D|+|V_M|} x_{ip} = 1 \qquad \forall i = 1,\ldots,|V_D| \\
&\sum_{i=1}^{|V_D|} x_{ip} \le 1 \qquad \forall p = 1,\ldots,|V_M| + |V_D|; \quad x_{ip} \in \{0,1\}.
\end{aligned}
\tag{10}
$$

[*] The detection probability can be made a function of nodes or edges.

Taking negative logarithm and neglecting the constant terms, Eq. (8) is rewritten as:

$$\begin{aligned}X^* = \arg\min_X & \sum_{i=1}^{|V_D|}\sum_{p=1}^{|V_M|} x_{ip}\left(-\log p(\underline{a}_D^{(i)} \mid \underline{a}_M^{(p)}) + \log\frac{1}{\Phi_1} + \log\left(\frac{1-P_d}{P_d}\right)\right) \\ & + \sum_{i=1}^{|V_D|}\sum_{p=1}^{|V_M|}\sum_{j\in N^+(i)}\sum_{q\in N^+(p)} x_{ip}x_{jq}\left(-\log p(\underline{a}_D^{\langle i,j\rangle} \mid \underline{a}_M^{\langle p,q\rangle}) + \log\frac{1}{\Phi_2} + \log\left(\frac{1-P_d}{P_d}\right)\right).\end{aligned} \tag{11}$$

2.3. Problem Formulation as QAP

Assuming, without loss of generality, that $|V_M| < |V_D|$, Eq. (11) can be rewritten in the following form:

$$\begin{aligned}X^* &= \arg\min_X \sum_{i=1}^{|V_D|+|V_M|}\sum_{p=1}^{|V_M|} x_{ip}\alpha_{ip} + \sum_{i=1}^{|V_D|+|V_M|}\sum_{p=1}^{|V_M|}\sum_{j=1}^{|V_D|+|V_M|}\sum_{q=1}^{|V_M|} x_{ip}x_{jq}\beta_{ijpq} \\ &= \arg\min_X \sum_{i=1}^{|V_D|+|V_M|}\sum_{p=1}^{|V_M|} x_{ip}\alpha_{ip}\cdot\frac{1}{|V_M|}\sum_{j=1}^{|V_D|+|V_M|}\sum_{q=1}^{|V_M|} x_{jq} + \sum_{i=1}^{|V_D|+|V_M|}\sum_{p=1}^{|V_M|}\sum_{j=1}^{|V_D|+|V_M|}\sum_{q=1}^{|V_M|} x_{ip}x_{jq}\beta_{ijpq} \\ &= \arg\min_X \sum_{i=1}^{|V_D|+|V_M|}\sum_{p=1}^{|V_M|}\sum_{j=1}^{|V_D|+|V_M|}\sum_{q=1}^{|V_M|} \frac{1}{|V_M|}x_{jq}x_{ip}\alpha_{ip} + \sum_{i=1}^{|V_D|+|V_M|}\sum_{p=1}^{|V_M|}\sum_{j=1}^{|V_D|+|V_M|}\sum_{q=1}^{|V_M|} x_{ip}x_{jq}\beta_{ijpq} \\ &= \arg\min_X \sum_{i=1}^{|V_D|+|V_M|}\sum_{p=1}^{|V_M|}\sum_{j=1}^{|V_D|+|V_M|}\sum_{q=1}^{|V_M|} \left(\frac{1}{|V_M|}\alpha_{ip} + \beta_{ijpq}\right)x_{ip}x_{jq}\end{aligned} \tag{12}$$

where

$$\alpha_{ip} = \begin{cases} -\log p(\underline{a}_D^{(i)} \mid \underline{a}_M^{(p)}) + \log\frac{1}{\Phi_1} + \log\left(\frac{1-P_d}{P_d}\right) & \forall 1 \le i \le |V_D|, \quad 1 \le p \le |V_M| \\ 0 & \text{otherwise} \end{cases}$$

$$\beta_{ijpq} = \begin{cases} -\log p(\underline{a}_D^{\langle i,j\rangle} \mid \underline{a}_M^{\langle p,q\rangle}) + \log\frac{1}{\Phi_2} + \log\left(\frac{1-P_d}{P_d}\right) & \forall 1 \le i \le |V_D|, \quad 1 \le p \le |V_M|, \quad j \in N^+(i), \quad q \in N^+(p) \\ 0 & \text{otherwise} \end{cases}.$$

Let $n_1 = \min(|V_M|,|V_D|)$, $n_2 = |V_M| + |V_D|$, Eq. (12) can be formulated as an asymmetric QAP:

$$\begin{aligned}X^* &= \arg\min_X \sum_{i=1}^{n_2}\sum_{p=1}^{n_1}\sum_{j=1}^{n_2}\sum_{q=1}^{n_1}\left(\frac{1}{n_1}\alpha_{ip} + \beta_{ijpq}\right)x_{ip}x_{jq} \\ \text{s.t.} &\quad \sum_{i=1}^{n_2} x_{ip} = 1; \quad \sum_{p=1}^{n_1} x_{ip} \le 1; \quad x_{ip} \in \{0,1\}.\end{aligned} \tag{13}$$

We augment the dimension to $n \times n$, $n = n_2$ with the added cost elements set to zero and convert it to a standard QAP of the form:

$$\begin{aligned}X^* &= \arg\min_X \sum_{i=1}^{n}\sum_{p=1}^{n}\sum_{j=1}^{n}\sum_{q=1}^{n} w_{ijpq}x_{ip}x_{jq} \\ \text{s.t.} &\quad \sum_{i=1}^{n} x_{ip} = 1; \quad \sum_{p=1}^{n} x_{ip} = 1; \quad x_{ip} \in \{0,1\}\end{aligned} \tag{14}$$

where

$$w_{ijpq} = \begin{cases} \frac{1}{n_1}\alpha_{ip} + \beta_{ijpq} & \text{if} \quad 1 \le i, \quad j \le n_2, \quad 1 \le p, \quad q \le n_1 \\ 0 & \text{otherwise} \end{cases}.$$

3. SOLUTION APPROACHES

The problem posed in (14) is a QAP, which has a broad range of applications requiring optimization and has been under intensive research since the 1950s. The QAP was first formulated by Tjalling C. Koopmans and Martin Beckman in the context of facility location [30]. Since then, it has received increasing attention from researchers in such diverse areas as economic activities, strategic planning, industrial design, statistical analysis, chemical reaction analysis, and numerical analysis [32]. Additionally, some standard and computationally intractable optimization problems, such as the Traveling Salesman Problem (TSP), maximum clique and the graph matching problem, are also known to be special cases of QAP [33]. Consequently, the development of computationally efficient algorithms for QAP will result in substantial advances in the aforementioned application areas. For examples, clinics can be located at the appropriate sites in an urban setting to handle emergencies; designers can reduce the cost when wiring the computer backboard; and higher accuracy can be obtained in the classification of objects [10].

3.1. Review of Relevant QAP Algorithms

QAP is known to be a NP-hard problem [19]. Generally, the algorithms for solving QAP can be classified into two categories, exact and heuristic. The first category of algorithms employ exhaustive search to obtain an optimal solution. The most frequently used exact algorithms are the branch-and-bound and dynamic programming. Branch-and-bound, often coupled with cutting plane methods [8], employs lower and upper bounds on the objective function to prune the nodes of the search graph [42]. Dynamic programming is employed for some special cases of QAP, specifically for tree QAPs [47]. Using these techniques, QAP instances of size less than 30 can be solved in a reasonable time [27]. Consequently, exhaustive search over the solution space is impractical for real-world problems. The second category of algorithms employs approximations (heuristics); these algorithms seek to generate near-optimal solutions for the QAP. Specifically, an important branch of heuristic algorithms, known as 'Meta-Heuristics,' provides a general approach to a wide range of intractable combinatorial optimization problems. In this paper, we will discuss and analyze Graduated Assignment Algorithm (GradAA) [22], Chaotic Tabu Search (Chaotic TS) [27], and Simulated Annealing (SA) [53] and compare them to m-best soft assignment algorithm (m-Best SAA) [25].

1) *Simulated Annealing*: Simulated Annealing (SA) is a probabilistic method for finding the global minimum of an objective function in large search spaces. The name derives from an annealing technique used in metallurgy, where heating and controlled cooling of a material reduces defects. Suppose X is the feasible solution of an optimization problem with the objective function f. Let $N(X)$ denote the set of neighborhood solutions of X. Every solution $X' \in N(X)$ can be reached directly from X via a 'move.' The difference in the objective function between solution X and X' is evaluated, i.e., $\Delta f = f(X') - f(X)$. If the 'move' improves the objective function, the new solution X' is accepted and saved as the current solution; otherwise, the 'move' will be accepted with the probability:

$$P(X,X') = e^{-|\Delta f|/T}. \tag{15}$$

Here, T is the current temperature, which will be progressively decreased until convergence is reached. The annealing operations are continued until a termination condition is satisfied.

There are three features that characterize an SA implementation: Neighborhood search, Annealing Schedule, and Termination Condition. Neighborhood search specifies a strategy for generating a new solution X' derived from the current solution X. For QAP, any neighborhood solution can be reached from X by interchanging two elements of the corresponding permutation. Annealing schedule involves selecting an initial temperature (typically large), progressively reducing this temperature during the search process, and invoking a test to detect equilibrium (convergence). The initial temperature is related to the maximal difference in the objective function value between any two neighborhood solutions as can be seen from Eq. (15). However, accurate computation of the maximal difference among neighborhood solutions is time consuming. Instead, various approximations are used [36]. The search process is terminated after a specified number of annealing iterations or when the objective function does not show improvement.

2) *Chaotic Tabu Search*: A chaotic Tabu search algorithm for QAP is proposed in [27]. In this algorithm, QAP is formulated as follows:

$$\min \sum_{i,j=1}^{n} a_{ij} b_{\underline{\pi}(i)\underline{\pi}(j)} \tag{16}$$

where $\underline{\pi}$ denotes the permutation of indices. Here another permutation $\underline{\pi}^{-1}$ is defined as the inverse function of $\underline{\pi}$ (if $i = \underline{\pi}(j)$, $j = \underline{\pi}^{-1}(i)$). A simple pair-wise exchange procedure is utilized to generate a new solution from the current one. The Tabu list is constructed such that it prohibits certain exchange moves. If an assignment (i,j) is in the Tabu list, $(\underline{\pi}(i), \underline{\pi}^{-1}(j))$ is also forbidden. A chaotic dynamic mechanism is applied to decide whether to keep the updated permutation. The Tabu search is implemented using a neural network. The output of neuron $\chi_{ij}(t)$ controls whether the exchange will be accepted or not: if $\chi_{ij}(t) > 0.5$, the permutation will be accepted. The output of neuron is calculated as follows:

$$\begin{aligned}
\xi_{ij}(t+1) &= \beta\Delta_{ij}(t)\\
\eta_{ij}(t+1) &= -W\sum_{k=1}^{n}\sum_{l=1,(k\neq i\vee l\neq j)}^{n}\chi_{kl}(t) + W\\
\gamma_{ij}(t+1) &= -\alpha\sum_{d=0}^{s-1}k_r^d\{\chi_{\underline{\pi}(j)\underline{\pi}^{-1}(i)}(t-d) + z_{\underline{\pi}(j)\underline{\pi}^{-1}(i)}(t-d)\} + \theta\\
\zeta_{ij}(t+1) &= -\alpha\sum_{d=0}^{s-1}k_r^d\{\chi_{ij}(t-d) + z_{ij}(t-d)\} + \theta \qquad (17)\\
x_{ij}(t+1) &= f\{\xi_{ij}(t+1) + \eta_{ij}(t+1) + \gamma_{ij}(t+1) + \zeta_{ij}(t+1)\}\\
&= \frac{1}{1+\exp\left(-\frac{1}{\varepsilon}(\xi_{ij}(t+1)+\eta_{ij}(t+1) + \gamma_{ij}(t+1)+\zeta_{ij}(t+1))\right)}
\end{aligned}$$

where $\Delta_{ij}(t)$ is the gain in the objective function value after exchanging elements i and j; β is a scaling parameter for $\Delta_{ij}(t)$; W is the connection weights; θ is the positive bias; k_r is the decay parameter for the Tabu effect; α is the scaling parameter for the Tabu effect; $z_{\underline{\pi}(j)\underline{\pi}^{-1}(i)}(t)$ is the accumulated output value $\chi_{ij}(t)$; $\eta_{ij}(t+1)$, $\gamma_{ij}(t+1)$ and $\zeta_{ij}(t+1)$ are internal states corresponding to the gain effect, and the two Tabu effects of exchanging (i, j) and $(\pi(i), \pi^{-1}(j))$, respectively.

3) *Graduated Assignment Algorithm*: The key ingredients of a Graduated Assignment Algorithm are deterministic annealing to avoid local optimum, iterative projective scaling to guarantee that assignment constraints are satisfied, and sparsity exploitation for efficient implementation.

Consider a general QAP given by

$$\min \sum_{i,j=1}^{n} \sum_{p,q=1}^{n} w_{ijpq} x_{ip} x_{jq}. \tag{18}$$

The Graduated Assignment Algorithm converts discrete QAP to a continuous one to avoid getting trapped by a local optimum. To formulate the idea, consider the simple problem of finding the maximum element within a set of variables $\{Q_i\}_{i=1}^{I}$. Define binary variables $m_i \in \{0,1\}$ such that $\sum_{i=1}^{I} m_i = 1$. This problem can be formulated as one of maximizing $\sum_{i=1}^{I} m_i Q_i$ subject to $\sum_{i=1}^{I} m_i = 1$ and $m_i \in \{0,1\}$. This discrete problem is converted into a continuous one by adding a controllable parameter β ($\beta > 0$) [22] and setting:

$$m_j = \frac{\exp(\beta Q_j)}{\sum_{i=1}^{I} \exp(\beta Q_i)}. \tag{19}$$

The use of exponentiation (the so-called *softmax*) ensures that the set $\{m_i\}$ has positive elements in the range $(0,1)$ and that they sum to unity. As β increases, the m_i corresponding to the maximal element in the set $\{Q_k\}_{k=1}^{I}$ converges to 1, while the rest of the elements in the set $\{\{m_k\}_{k=1;k\neq i}^{I}\}$ converge to 0. In the context of Graduated Assignment Algorithm (GradAA), the value of the control parameter β is progressively increased to force the continuous values closer to the discrete counterpart. Thus, deterministic annealing is a key ingredient of GradAA.

Iterative projective scaling is a process that can transfer any nonnegative square matrix into a doubly stochastic matrix by normalizing the rows and columns in the matrix. The Graduated Assignment Algorithm adopts this strategy, along with deterministic annealing, to solve the general assignment problem.

The basic idea of Graduated Assignment Algorithm in the context of QAP is to approximate the QAP by its Taylor series expansion around an initial assignment matrix X^0 as follows:

$$\sum_{i,j=1}^{n} \sum_{p,q=1}^{n} w_{ijpq} x_{ij} x_{pq} \approx \sum_{i,j=1}^{n} \sum_{p,q=1}^{n} w_{ijpq} x_{ij}^{0} x_{pq}^{0} + \sum_{i,j=1}^{n} Q_{ij}^{0} (x_{ij} - x_{ij}^{0}) \tag{20}$$

where

$$Q_{ij}^{0} = \left. \frac{\partial \sum_{i,j=1}^{n} \sum_{p,q=1}^{n} w_{ijpq} x_{ij} x_{pq}}{\partial x_{ij}} \right|_{X=X^0} = 2 \sum_{p,q=1}^{n} w_{ijpq} x_{pq}^{0}.$$

Therefore, solving QAP is equivalent to solving a succession of assignment problems. In GradAA, a probabilistic solution is generated for the linearized QAP for use in the next iteration. To deal with ties, a heuristic method is applied at the end of the algorithm to convert the doubly stochastic matrix to a permutation matrix.

3.2. Soft Assignment via m-best Assignment Algorithm

For a general optimization problem, the optimal solution can be obtained by searching among a number of local optimum points. In this vein, there are two interrelated issues: one is to quickly find a local optimum and the other is to jump from one local optimum to another. These two issues are termed *intensification* and *diversification*, respectively. Intensification means optimizing the objective function's value by seeking solutions that are in the neighborhood of a local optimum. Diversification implies moving from one local optimum's region to another in order to avoid getting trapped at a local optimum that is not a global one.

Our m-Best soft Assignment Algorithm for solving the QAP involves the following steps: (i) apply m-Best soft 2-D assignment procedure to quickly generate a solution that is close to a local optimum (intensification); (ii) employ local search (Genetic Algorithm) to obtain a near-optimal solution; and (iii) repeat (i) and (ii) with different initial assignments, i.e., employ multistart strategy for diversification.

Similar to a Graduated Assignment Algorithm, the key idea of m-Best soft search procedure is to solve the QAP by solving a series of linearized QAPs. We adopt a linearized form proposed in [51]. Supposing X^{k-1} is known at kth iteration. Using the fact that $x^2 = x$ for binary variables, the QAP can be approximated by:

$$\min_{X} \sum_{u,v=1}^{n} \left\{ w_{uvuv} + 0.5 \sum_{\substack{i,j=1 \\ i\neq u \\ j\neq v}}^{n} [w_{ijuv} + w_{uvij}] x_{ij}^{(k-1)} \right\} x_{uv}^{(k)} \tag{21}$$

$$\text{s.t.} \quad \sum_{u=1}^{n} x_{uv}^{(k)} = 1, \quad \sum_{v=1}^{n} x_{uv}^{(k)} = 1, \quad x_{uv}^{(k)} \in \{0,1\}.$$

Different from the Graduated Assignment Algorithm, the Linearized Assignment Problem is solved using the auction algorithm or the JVC algorithm. These

TABLE I
m-Best Assignment Algorithm for the QAP

```
1) Initialize max_iteration and max_loops. Set loop = 1.
2) Initialize $\hat{X}$ to a uniform matrix.
3) Calculate the modified cost matrix
       $$\hat{w}_{pq} = w_{pqpq} + \frac{1}{2}\sum_{\substack{i,j=1\\ i\neq p\\ j\neq q}}^{n}[w_{ijpq} + w_{pqij}]\hat{x}_{ij}^{(k-1)}.$$
4) Solve the assignment problem to obtain m-best solutions
   $\{X_d, d = 1,\ldots,m\}$ with the corresponding cost values
   $\{c(d), d = 1,\ldots,m\}$.
5) Compute $\hat{X} = \sum_{d=1}^{m}\lambda(d)X_d$, where
       $$\lambda(d) = \frac{\exp[-c(d)/c(m)]}{\sum_{l=1}^{m}\exp[-c(l)/c(m)]}.$$
6) Check whether the $\{X_d, d = 1,\ldots,m\}$ have converged. If not,
   go to step 3. Otherwise, go to step 7.
7) Employ genetic algorithm to search for a better feasible
   solution $X^*$ using the top solutions from the m-best
   assignments and the best feasible solution found so far.
8) Make $\hat{X} = \hat{X} - \mu X^*$ $\mu = 1/n(0.5)^{loop}$, $loop = loop + 1$ and
   check whether the max_loops reached. If not, go to step 3.
```

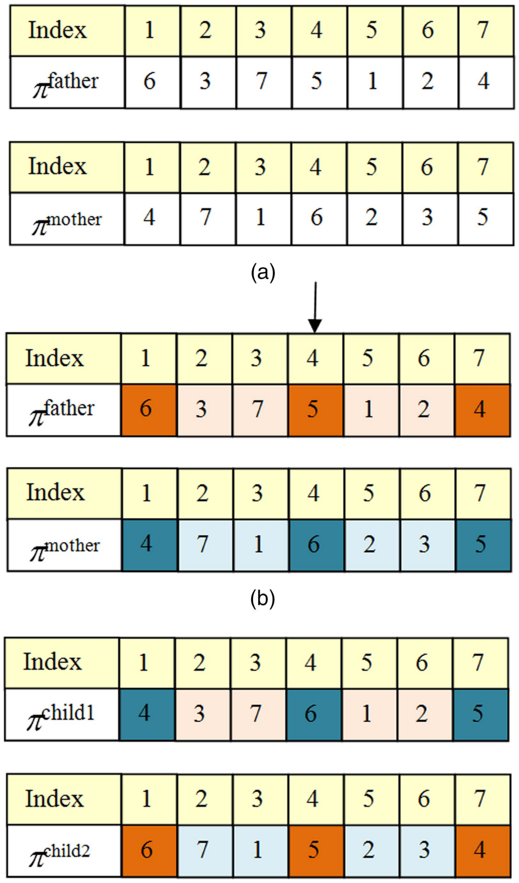
algorithms have proven to be among the best for solving the assignment problems [2]. The soft assignment matrix is calculated as a convex combination of the *m*-Best assignment matrices as follows:

$$\hat{X} = \sum_{d=1}^{m}\lambda(d)X_d \qquad \lambda(d) = \frac{\exp[-c(d)/c(m)]}{\sum_{l=1}^{m}\exp[-c(l)/c(m)]} \tag{22}$$

where X_d is the *d*th best solution to the assignment problem with a corresponding cost $c(d)$. This soft assignment matrix will lead to a more directed search because only a cluster of good solutions are saved to guide the next iteration's search. Fast convergence is achieved by this intensive search strategy. A heuristic is needed to convert soft assignments into feasible (0–1) assignments. A simple elitism-based genetic algorithm is employed to seek a feasible solution based on soft assignments.

We also implemented a diversification strategy based on a multi-start method. The soft assignment matrix specifies the next iteration's cost matrix for the 2-D assignment problem, while the probability that each of the *m* assignment matrices being optimal is determined by their corresponding assignment costs. Initially, the soft assignment matrix is set up as uniform to guarantee that each element in the assignment matrix *X* has equal probability of being 0 or 1. However, in order to explore the search space (i.e., diversification), we move away from the best feasible solution found via the genetic search (step 8 below).

A genetic local search is proposed for improving the solution. The design of genetic local search includes the way the chromosomes are encoded and decoded to represent the binary assignment matrix, the crossover operator and mutation operator to generate the feasible solutions, and the strategy to manage the population.



Index	1	2	3	4	5	6	7
π^{father}	6	3	7	5	1	2	4

Index	1	2	3	4	5	6	7
π^{mother}	4	7	1	6	2	3	5

(a)

Index	1	2	3	4	5	6	7
π^{father}	6	3	7	5	1	2	4

Index	1	2	3	4	5	6	7
π^{mother}	4	7	1	6	2	3	5

(b)

Index	1	2	3	4	5	6	7
π^{child1}	4	3	7	6	1	2	5

Index	1	2	3	4	5	6	7
π^{child2}	6	7	1	5	2	3	4

(c)

Fig. 3. Illustrative example of crossover.

1) Chromosome Representation: A permutation vector $\underline{\pi} = [\pi_1, \pi_2, \ldots, \pi_n]$, where $\pi_i = p$ iff $x_{ip} = 1$, is adopted to describe the node-to-node association relationship.

2) Crossover and Mutation Operators: The crossover operator involves exchanging parents' genes to reproduce the offspring while maintaining assignment feasibility. Suppose two parents' chromosomes are specified as $\underline{\pi}^{\text{father}} = [\pi_1^0, \pi_2^0, \ldots, \pi_n^0]$ and $\underline{\pi}^{\text{mother}} = [\pi_1^1, \pi_2^1, \ldots, \pi_n^1]$. Firstly, we remove the elements such that $\pi_i^0 = \pi_i^1$, and obtain $[\pi_{i1}^0, \pi_{i2}^0, \ldots, \pi_{in'}^0]$ and $[\pi_{j1}^1, \pi_{j2}^1, \ldots, \pi_{jn'}^1]$, as shown in the example of Fig. 3(a). Secondly, we start from a random index $i \in \{i_1, i_2, \ldots, i_{n'}\}$, and search for $j \in \{j_1, j_2, \ldots, j_{n'}\}$ such that $\underline{\pi}^{\text{mother}}(j) = \pi_j^1 = \pi^{\text{father}}(i) = \pi_i^0$, save index i in set η. Next, set $i = j$ and repeat the second step until j is already in the set η. For the example in Fig. 3(b), random index = 4, and set $\eta = \{4, 1, 7\}$. Now, we can swap the chromosomes whose indices are contained in set η to generate feasible solutions. The mutation process refers to increasing the chromosomes' diversity by introducing random variations. The mutation operator we employed is implemented by exchanging two randomly selected chromosomes of a feasible

solution. In the network identification problem, this implies the exchange of nodes in data/model network with the corresponding associated nodes in the model/data network subject to the 1-to-1 assignment constraints.

3) Population Management: Population management deals with how many child solutions are generated at each generation and which of these solutions will lead to the next generation's solutions. We generate *max_population* child solutions at each generation and then pick the top *max_population* members from the current population (*max_population* child solutions plus *max_population* parent solutions) to produce the next generation. An elitist strategy is adopted and the selection probability for crossover is inversely proportional to the candidate solution's QAP cost [23], [40].

4. EXPERIMENTAL RESULTS

In this section, we investigate the performance of *m*-Best SAA, Graduated Assignment Algorithm (GradAA), Chaotic Tabu Search (Chaotic TS) and Simulated Annealing (SA) in the context of the organizational identification problem. In addition, we also apply *m*-Best SAA to general QAP from a standard QAP library, and compare its solutions to the hitherto best solutions found for these problems in the literature.

4.1. Application to Network Identification Problem

The command and control organization is a collection of command and control nodes and resources connected via command, control, communication, and mission structures. The roles, responsibilities, and relationships among C^2 nodes and resources constrain how the organization is able to operate. Here, C^2 nodes are entities with information-processing, decision-making, and operational capabilities that can control the necessary units and resources to execute mission tasks, provided that such an execution does not violate the concomitant capability thresholds (e.g., limited weapon supplies, fixed communication bandwidth, bounded human information-processing capabilities [21]). A C^2 node can represent a single commander, a liaison officer, a system operator, or a command cell with its staff. A set of physical platforms and assets, C^2 nodes, and/or personnel can be aggregated to a resource (e.g., bomber maker, truck, weapons system, etc.). A resource is considered as a physical asset of an organization that provides resource capabilities and is used to execute tasks. The roles and responsibilities of the C^2 nodes and resources identify possible operational and tactical policies, viz., the decisions they can make, and possible actions they can perform.

Scenario & Hypothesis Space

The adversary organization of interest is illustrated in Fig. 4, which is comprised of Decision Makers (Black, Blue and etc. in Fig. 4(b)), platforms (BMT, IT and etc. in Fig. 4(c)) and resources (Transport, Strike and etc. in Fig. 4(a)). The top commander (Black, Fig. 4(b)) sets the initial conditions and provides the overall intent for an operation. Four sub-commanders, blue, green, brown and red, distribute the responsibilities among lower-level units (platforms), and coordinate these seemingly independent activities to achieve the mission objectives (Fig. 4(c)). The fundamental need for communications (Fig. 4(d)) significantly constrains the options for both command and control, making communications infrastructure a critical feature of a C^2 system.

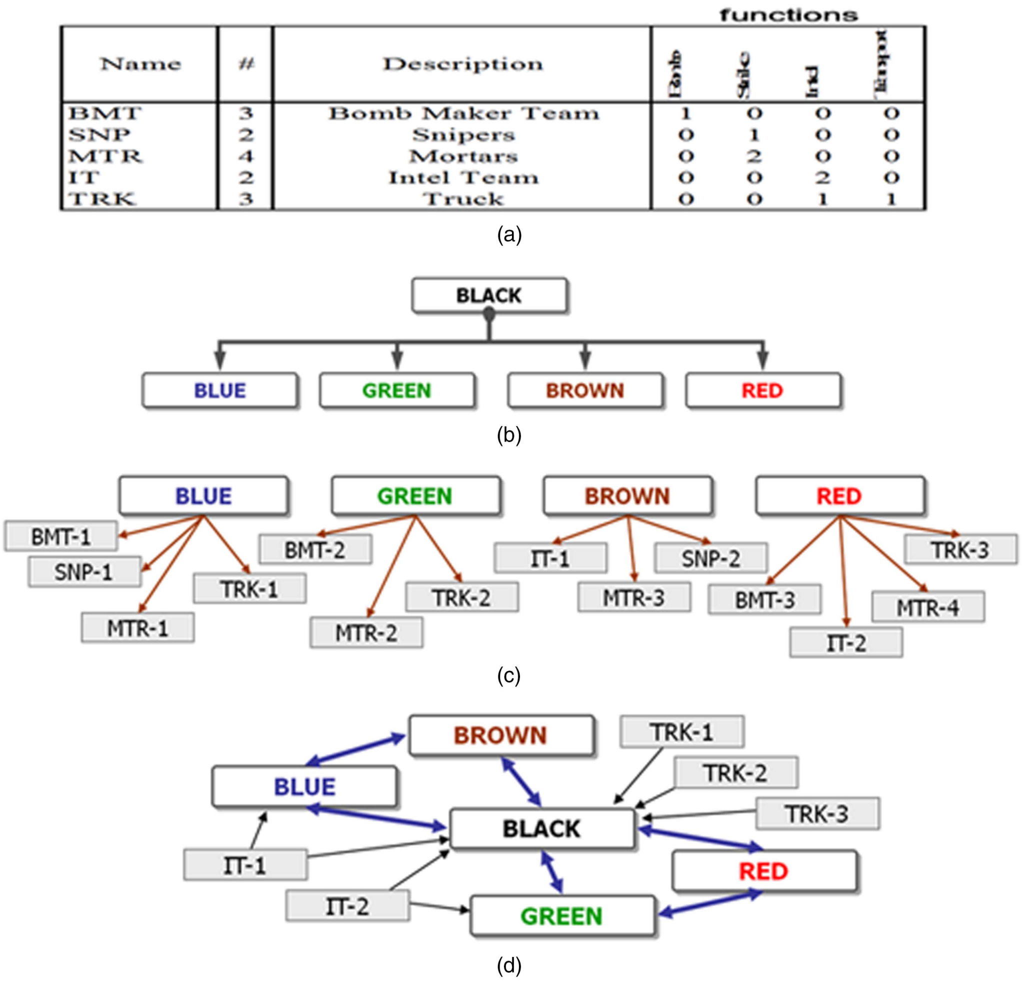
TABLE II
The Hypothesis Space

Hypothesis Space No.	H1	H2	H3	H4	H5	H6	H7
Structure Type	*F*	*D*	*F*	*D*	*I*	*F*	*D*

Seven hypothesized model networks are provided, which are categorized into 3 classes of organizational structures, Functional organization (F), Divisional organization (D) and a hybrid of the two (I). The D organization and F organization are two extreme cases of organizational structures considered here. In a D organization, a commander controls multiple types of assets and has general knowledge of these assets; the activities conducted by the members in this organization are restricted to a certain geographic area of responsibility. For example, one of the sub-commanders (blue, green, brown or red) may control a set of assets consisting of a bomb maker, an intelligence person and a transportation person (truck driver). On the other hand, a commander in an F organization controls a single asset type and has special knowledge on the asset the commander controls. Thus, the operations of an F organization cross multiple geographic regions. For example, a commander in an F organization is able to control one type of resource, such as one of intelligence, transportation, and attacking resources. An I organization has an organizational structure that is a hybrid of D and F organizations. The seven instances of hypothesized organizations are built within these three types of organizational structures, shown in Table II.

Observations & Data Processing

1) *Content of Observations*: We restrict our observations to several types of intelligence information that are currently feasible. We assume that the observations include the set of tracked (monitored) individuals whose positions in the organization we need to determine, information about these individuals, and identified adversary's resources. Tractable information regarding the individuals encompasses their attributes and resources—e.g., expertise of individuals, training, background, affiliation, cultural characteristics, family ties, etc. Information about adversary's resources may include detection of its military assets and their capabilities, communication patterns, political connections, and financial capa-



			functions			
Name	#	Description	Bomb	Strike	Intel	Transport
BMT	3	Bomb Maker Team	1	0	0	0
SNP	2	Snipers	0	1	0	0
MTR	4	Mortars	0	2	0	0
IT	2	Intel Team	0	0	2	0
TRK	3	Truck	0	0	1	1

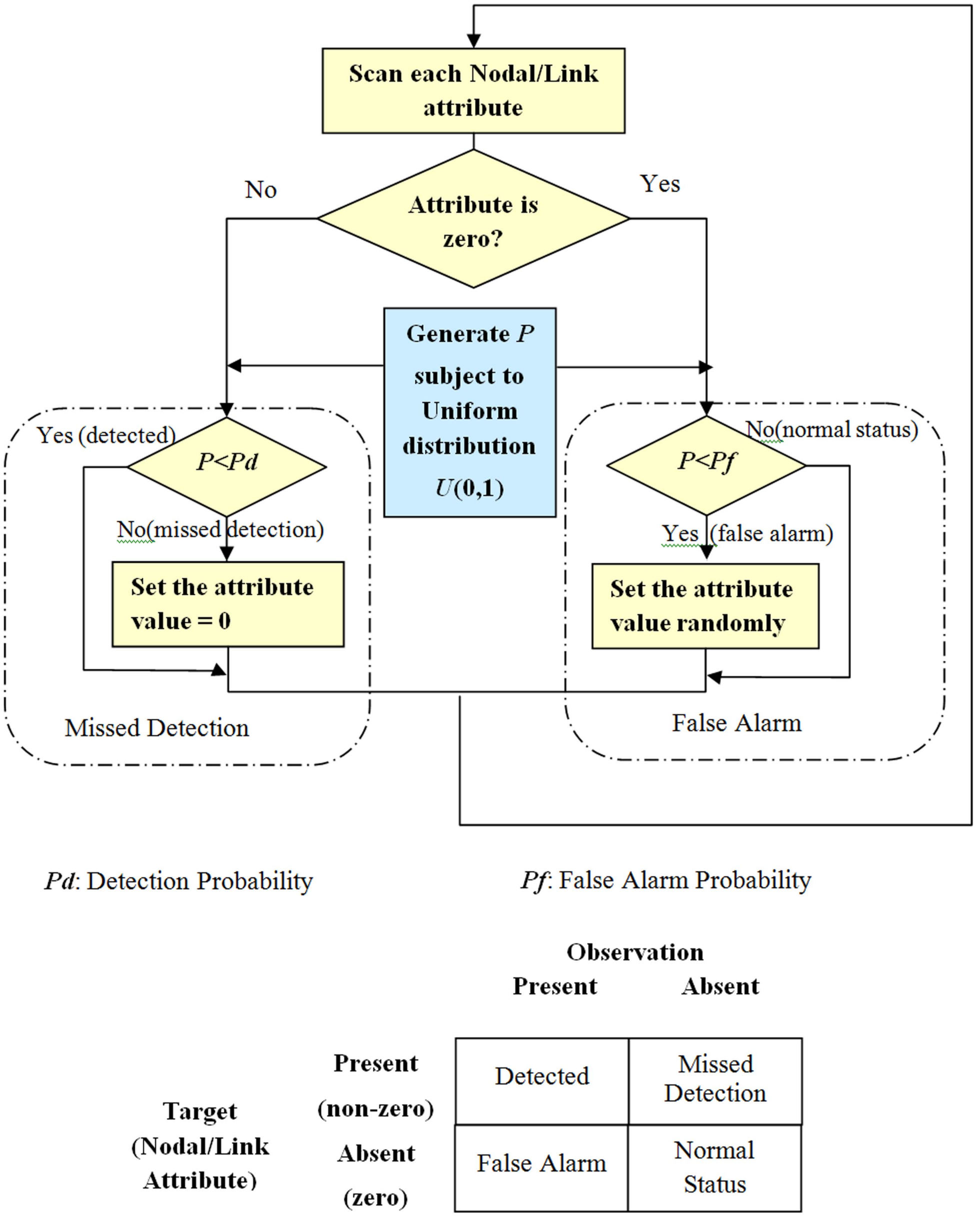
Fig. 4. Illustrative example of the experiment: (a) resources of C^2 organization (b) C^2 nodes & command structure (c) control structure (d) communication structure.

bilities. In addition, the observed information may also include transactions that involve these entities; these comprise partially deciphered communications among the individuals and the individuals' actions—the involvements in various observable activities. Communication observations may include some classification of communication content, e.g., request for or transfer of information, resources, actions, acknowledgements, directions, etc. Action observations may include functions/tasks performed, such as individuals committing the same crime, performing financial or business transactions, or using the resources in covert or open operations. Such data is typically very noisy and sparse due to challenges in data collection, e.g., limited sensors and/or human intelligence, security of adversary communication networks, uncertainty in message translation, data association uncertainty, attempts of the adversary to conduct deceptive actions, etc.

2) *Error Model & Data Fogging*: Generally, the noise in the observations can be categorized as: missing data (miss), deceptions (false alarms), and corrupt messages (mislabels) (as in Fig. 2). In this experiment, we consider the errors from the first two categories. The noisy observations are built from the true hypothesized network by deleting and adding observations with various degrees of uncertainty (high, medium, and low), which is termed the data fogging process [12].

To illustrate the fogging process, let us define a single data point as an observation on a specific node/link attribute. A noisy model is developed to generate the observed data network as indicated in Fig. 5. If the node/link attribute is zero, meaning that the entity (commander, platform or resource) does not have any capability or activity for the specific attribute, we probabilistically generate a false alarm for this attribute.

For each type of organization, the classes of nodes, messages, tasks, areas of responsibility and messages among different nodes are provided. In order to facilitate the simulation and fit the data to our model, we populate a database of node and link attributes, respectively. The nodes are divided into three hierarchical levels: commander (decision maker DM) level, platform (aggregation of a set of resources) level, and resource level. The attributes of nodes at the three levels are stored in three tables in a database. For nodes at the DM level, the attributes are control capabilities of platforms; for nodes at the platform level, the attributes are resource capabilities and operational areas; finally, for the resource


Scan each Nodal/Link attribute
Attribute is zero?
No
Yes
Generate P subject to Uniform distribution $U(0,1)$
Yes (detected)
$P<Pd$
No(missed detection)
Set the attribute value = 0
Missed Detection
No(normal status)
$P<Pf$
Yes (false alarm)
Set the attribute value randomly
False Alarm
Pd: Detection Probability
Pf: False Alarm Probability

		Observation	
		Present	Absent
Target (Nodal/Link Attribute)	Present (non-zero)	Detected	Missed Detection
	Absent (zero)	False Alarm	Normal Status

Fig. 5. Data fogging process.

nodes, the attributes are task execution capabilities and operational areas. The attributes for links among these nodes are the message types.

The categories of noise that are used to generate observation data in this experiment is shown in Table III. There are 12 noise categories of noise uncertainty: Low-1 to Low-4, Med-1 to Med-4 and High-1 to High-4. We varied the uncertainty from very low level (5% missing data, 5% deceptions) to a very high level (60% data missing and 60% deceptions).

Simulation Results

1) *Algorithm Performance Evaluation*: We first investigate the accuracy and computational efficiency of network matching algorithms on 42 organizational structure identification problems. The 42 problems were created by selecting 6 randomly generated data networks (Observations) to be associated with 7 model networks (Hypotheses). Besides *m*-Best Soft Assignment Algorithm (*m*-Best SAA), we have also experimented with the Graduated Assignment Algorithm (GradAA), Chaotic Tabu Search (Chaotic TS), Simulated Annealing Algorithm (SA). Fig. 6 shows the matching accuracies and CPU times for the 42 structure identification problems considered here. It is evident that the *m*-Best SAA achieves the minimal objective function value for all the 42 problems, while simulated annealing is off by 2.15% from the optimal, GradAA by 3.64%, Chaotic TS by 2.8%. As shown in Fig. 6, we also note that both GradAA and Chaotic TS produce suboptimal results, although they are competitive computationally. Specifically, the average CPU time for Chaotic TS is 1.3865

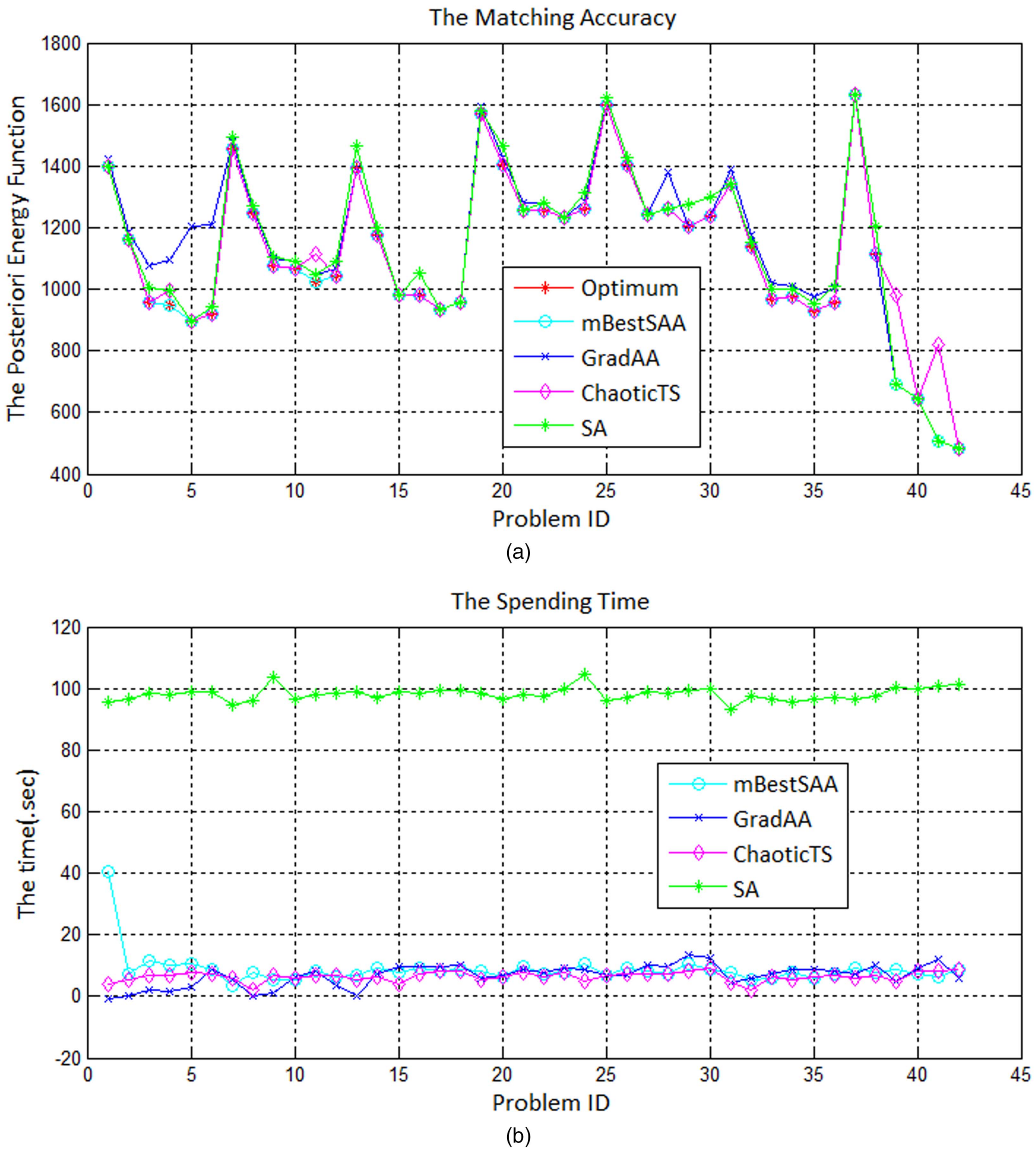


Fig. 6. Algorithm performances: (a) comparison of organization matching accuracy (b) comparison of computation time.

TABLE III
The Categorized Noisy Data

Noise Category	Missing Data (%)	Deceptions (%)
Low-1	5	5
Low-2	10	10
Low-3	15	15
Low-4	20	20
Med-1	25	25
Med-2	30	30
Med-3	35	35
Med-4	40	40
High-1	45	45
High-2	50	50
High-3	55	55
High-4	60	60

sec., while that for GradAA it is 1.4264 sec. and for *m*-Best SAA it is 1.6355 sec. The simulated annealing algorithm is the slowest with a CPU time of 136.218 sec. Based on its superior accuracy and modest computational requirements, *m*-Best SAA is a computational scheme of choice for our organizational identification problem.

2) *Sensitivity Analysis*: Next, we considered the base-line organization structures (H1–H7) and generated data networks from them according to the noise categories shown in Table III. We perform network matching between the observed data and the hypothesized organizations. The hypothesis with the least posterior energy from the graph matching algorithm is picked as the identified organization, shown in Table IV. Each row represents the base-line organization structure from H1 to H7 and each column represents the noise level from Low-1 to High-4. We also show the results of the identified organization type (F, D or I) in Table V.

From the table, we note that when noise levels are Low ($\leq 20\%$ missing data and $\leq 20\%$ deceptions), the

TABLE IV
The Identified Organization Structures

	Low-1	Low-2	Low-3	Low-4	Med-1	Med-2	Med-3	Med-4	High-1	High-2	High-3	High-4
H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H7
H2	H2	H2	H2	H2	H2	H2	H2	H2	H2	H2	H2	H2
H3	H3	H3	H3	H3	H3	H3	H7	H3	H3	H1	H7	H3
H4	H4	H4	H4	H7	H4	H4	H4	H4	H4	H2	H1	H2
H5	H5	H5	H5	H5	H5	H5	H5	H7	H5	H7	H5	H7
H6	H6	H6	H6	H6	H6	H6	H6	H6	H6	H6	H2	H7
H7	H7	H7	H7	H7	H7	H7	H7	H7	H7	H7	H7	H7

TABLE V
The Identified Organization Types

	Low-1	Low-2	Low-3	Low-4	Med-1	Med-2	Med-3	Med-4	High-1	High-2	High-3	High-4
H1	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*D*
H2	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*
H3	*F*	*F*	*F*	*F*	*F*	*F*	*D*	*F*	*F*	*F*	*D*	*F*
H4	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*F*	*D*
H5	*I*	*I*	*I*	*I*	*I*	*I*	*I*	*D*	*I*	*D*	*I*	*D*
H6	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*D*	*D*
H7	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*

algorithm can recognize the correct organization (H1–H7) with an accuracy of 96.43% and can correctly identify the organization structure type (F, D or I) with 100% accuracy for all the cases. When the noise level increases to Medium (25–40% missing data and 25%–40% deceptions), the accuracy falls to 92.86%. As the noise level is increased to High (45–60% missing data and 45%–60% deceptions), the node/link attributes associated with specific commander/DM become very noisy, and the algorithm breaks down with accuracy of 64.29%. The Functional Organizational Structure, where each commander/DM controls single type of asset, has 100% correct identification in the presence of Low noise level, 91.67% at the Medium noise level, and is comparably difficult to be correctly identified (with only 58.33% accuracy) when the noise level is increased to High level. For the Divisional Organizational Structure, where each commander/DM controls multiple types of assets, our approach has relatively robust performance in that it achieves 91.67% correct identification at the Low noise level, and 100% for the Medium and 75% for High noise levels.

Fig. 7 depicts the Receiver Operating Characteristics (ROC) operating points for the seven hypothesized networks labeled by H1 through H7 for all the noise levels. As shown, all the hypothesized networks have $84.52 \pm 1.72\%$ average true positive rate and $2.58 \pm 0.17\%$ average false positive rate. The overall identification accuracy is defined as:

$$\mathrm{Acc}_d = \frac{\mathrm{TP} + \mathrm{TN}}{L}. \qquad (23)$$

TP (True Positives) denotes the number of correct recognitions that the observed networks are correctly recognized as the corresponding hypothesized networks. Here, TN (True Negatives) denotes the correct "negative" detections that the data networks are correctly identified as not originating from the observations of the hypothesized network, and L denotes the total number of observed data networks ($L = 84$). We list the accuracies for each hypothesized network in Table VI. Our approach achieves 95.58% average accuracy based on the noisy observed data networks.

4.2. Standard Quadratic Assignment Problem

In addition to the organizational identification problem, the m-Best SAA can be used to solve the general QAP as well. We conducted the following experiment using examples from the QAP library, available from the University of Copenhagen (http://www.opt.math.tu-graz.ac.at/karisch/qaplib and http://www.diku.dk/karisch/qaplib). It consists of 16 libraries with 131 problems, along with optimal solutions or hitherto best known solutions. In this section, we present the results on four of these libraries, specifically the solution accuracy and timing results for the four libraries. Our algorithm is implemented in MATLAB running on a 2.2 GHz PC with 2046 MB memory. These timings can be improved by a factor of ten or more by implementing the algorithm in a low-level language such as C. Here Lib denotes the name of library while Gap is defined as:

$$\mathrm{Gap} = \frac{\mathrm{Cost}_{m\mathrm{BestSAA}} - \mathrm{Cost}_{\mathrm{lib}}}{\mathrm{Cost}_{\mathrm{lib}}}. \qquad (24)$$

The problems in Lib 'Bur' [3] seek to minimize the overall typing-time given the knowledge of frequency of each pair of letters and average typing time. The result can be used to design a typewriter keyboard. A greedy randomized adaptive search procedure (GRASP) [32] approach, coded in FORTRAN, has an average time of

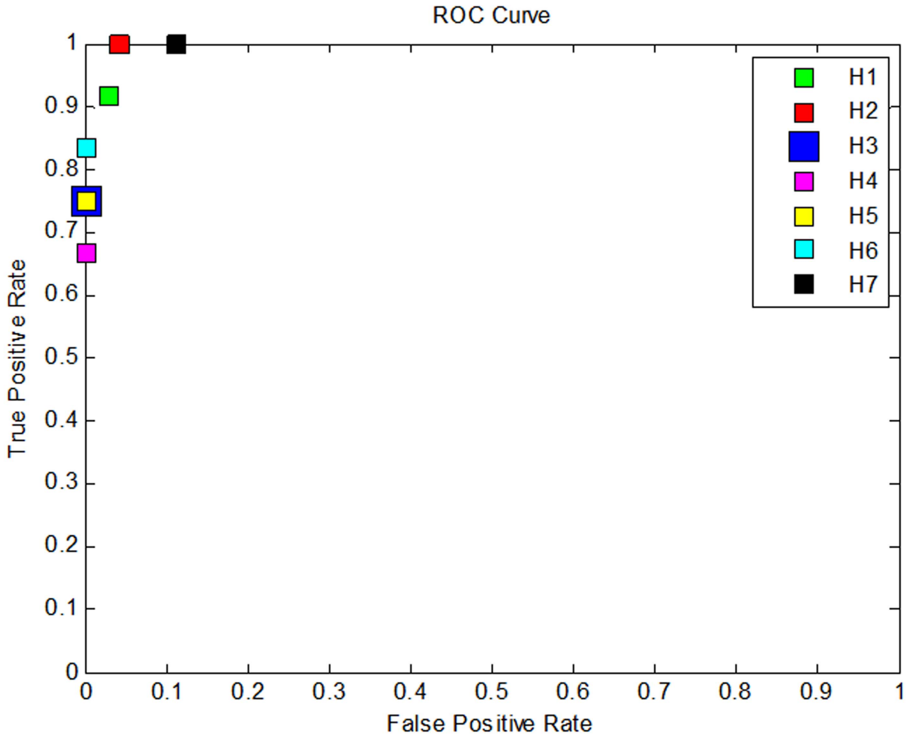


Fig. 7. Operating points on receiver operating characteristic curve.

TABLE VI
Identification Accuracies of Hypothesized Networks

	H1	H2	H3	H4	H5	H6	H7
Acc.	96.43%	96.43%	96.43%	95.24%	96.43%	97.62%	90.48%

TABLE VII
Four QAP Libs Considered

Library	Had	Nug	Bur	Chr
Average Gap	0%	0%	0.084%	1.58%

TABLE VIII
Accuracy and Computation Time on BUR Lib Problems

Problem	Bur26a	Bur26e	Bur26g	Bur26h
Gap to Known Best Solution	0.3%	0.66%	0.51%	0.15%
MATLAB Time (sec.)	1861.3	2499.6	1165.9	1226.8

TABLE IX
Computation Time on HAD and NUG Lib Problems

Problem	Nug12	Had12	Had14	Had16
MATLAB Time (sec.)	37.625	44.096	88.489	180.59

1664.3 sec. for solving the same size problem [35]. It is apparent that our algorithm can have a better tradeoff between time and degree of optimality.

Problems in Lib 'Had' [24] and 'Nug' [30] are assigning facilities to locations. The *m*-best soft assignment algorithm worked well on these problems as shown in Table IX. An improved heuristic algorithm is adopted for Nug12, with a CPU time of 3.43 seconds. In [24], a new bound using orthogonal relaxation is presented to obtain the lower bound for the 'Had' set of QAPs; no timings are available.

Problems in Lib 'Chr' [6] are special cases of QAPs called Tree QAPs. An exact algorithm based on dynamic programming is applied. We note that our algorithm works well for the Tree QAPs.

5. CONCLUSION

In this paper, we formulated a network identification problem using a Maximum Likelihood Ratio Criterion, coupled with an *m*-Best assignment algorithm for solving the resulting QAP. The model and the methodology enable the computation of an energy function of the hypothesized organizational structure and processes, given the observed behavior of members in the organization. The focus of the paper was on identifying the mappings between hypothesized nodes of an adversary command organization and tracked individuals and resources. The hypothesized organizations are predefined in the knowledge library according to available intelligence regarding similar adversary organizations, well-known structural forms from organizational theories, as well as specific existing structures that analysts propose. Our modeling framework and solution methodology have great potential to enhance the capabilities of discovering competitive organizations and adversary networks.

In this paper, the network identification problem is solved assuming that a library of possible model net-

TABLE X
Accuracy and Computation Time on CHR Lib Problems

Problem	Chr12a	Chr12b	Chr12c	Chr15a	Chr15b
Gap to Known Best Solution	0%	0%	0.27%	7.62%	0%
MATLAB Time by our algorithm (sec.)	33.177	49.493	51.021	81.343	272.878
Time in [6] (sec)	9.4	2.8	1.2	61.3	28.0

work structures is already available. However, the creation of a model network library is expensive in terms of human resources, time or economic cost. Moreover, the model networks need to be periodically updated because they may become irrelevant or change over time. In order to overcome these limitations, we are focusing on the following four extensions to this work: (1) Given a batch of N data networks and assuming that each of them has *a single model network* of *one type* embedded in it, learn the single model network that best matches the N data networks; (2) Given a batch of N data networks and assuming that each of them has *a single model network* of *not necessarily the same type* embedded in it, learn the model networks that best match the N data networks; (3) Given a batch of N data networks and assuming that each of them has *one or more model networks* embedded in it, learn the model networks that best match the N data networks; and (4) Given a *temporally evolving data network* and a set of learned model networks, identify and track active model networks over time. The latter problem involves a multi-dimensional QAP.

REFERENCES

[1] Y. Bar-Shalom, P. Willett, and X. Tian
Tracking and Data Fusion: A Handbook of Algorithms.
2011.

[2] D. Bertsekas
An auction algorithm for shortest paths.
Society for Industrial and Applied Mathematics Journal on Optimization, **1** (1991), 425–447.

[3] R. Burkard and E. C-Ela
Quadratic and three-dimensional assignment problems.
Annotated Bibliographies in Combinatorial Optimization, 1996.

[4] T. Caelli and T. Caetano
Graphical models for graph matching: Approximate models and optimal algorithms.
Pattern Recognition Letters, **26** (2005), 339–346.

[5] W. J. Chrismas, J. Kittler, and M. Petrou
Structural matching in computer vision using probabilistic relaxation.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **17**, 8 (Aug. 1995), 749–764.

[6] N. Christofides and E. Benavent
An exact algorithm for the quadratic assignment problem.
Operations Research, **37** (1980), 760–768.

[7] F. R. K. Chung
Spectral graph theory.
American Mathematic Society, CBMS Series, 1997.

[8] J. Clausen and M. Perregaard
Solving large quadratic assignment problems in parallel.
Computational Optimization and Applications, **8** (1997), 111–127.

[9] T. R. Coffman and S. E. Marcus
Dynamic classification of groups through social network analysis and HMMs.
In *The Proceedings of IEEE Aerospace Conference*, 2004.

[10] C. Commander
A survey of the quadratic assignment problem with applications.
Morehead Electronic Journal of Applicable Mathematics, 2003.

[11] M. Dombroski and K. Carley
Netest: estimating a terrorist network's structure.
Computational & Mathematical Organization Theory, **8** (2002), 235–241.

[12] E. Entin, R. Grier, G. Levchuk, and T. Jefferson
NETSTAR: An intelligence analyst's decision tool.
Aptima Inc., 2007.

[13] M. A. Eshera and K. S. Fu
An image understanding system using attributed symbolic representation and inexact graph-matching.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **8**, 5 (1986), 604–618.

[14] M. Fernandez and G. Valiente
A graph distance metric combining maximum common sufgraph and minimum common supergraph.
Pattern Recognition Letter, **22**, 6–7 (2001), 753–758.

[15] A. W. Finch, R. C. Wilson, and E. R. Hancock
An energy function and continues edit process for graph matching.
Neural Computation, **10**, 7 (Oct. 1998), 1873–94.

[16] O. Frank
Sampling and inference in a population graph.
International Statistical Review, **48** (1980), 33–41.

[17] O. Frank
Network sampling and model fitting.
Models and Methods in Social Network Analysis, Cambridge University Press, 2005.

[18] L. C. Freeman
Centrality in social networks: Conceptual clarification.
Social Networks, (1979), 215–240.

[19] M. Garey and D. Johnson
Computers and Intractability: A Guide to the Theory of NP-Completeness.
New York: Freeman, 1979.

[20] L. Getoor
An introduction to probabilistic graphical models for relational data.
Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, **29**, 1 (2006), 32–39.

[21] T. Gilovich, D. Griffin, and D. Kahneman
Heuristics and Biases: The Psychology of Intuitive Judgment.
Cambridge University Press, 2002.

[22] S. Gold and A. Rangarajan
A graduated assignment algorithm for graph matching.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **18**, 4 (Apr. 1996), 377–388.

[23] D. Goldberg
Genetic Algorithms in Search, Optimization and Machine Learning.
Addison-Wesley, 1989.

[24] S. Hadley, F. Rendl, and H. Wolkowicz
A new lower bound via projection for the quadratic assignment problem.
Mathematics of Operations Research, **17** (1992), 727–739.

[25] X. Han, K. R. Pattipati, C. Park, and G. M. Levchuk
Organizational structure identification using a hidden Markov random field model and a novel algorithm for quadratic assignment problem.
IEEE International Conference on System, Man, and Cybernetics, Singapore, Oct. 2008.

[26] E. R. Hancock and J. Kittler
Discrete relaxation.
Pattern Recognition, **23** (1990), 711–733.

[27] M. Hasegawa, T. Ikeguchi, and K. Aihara
A novel chaotic search for quadratic assignment problems.
European Journal of Operational Research, **139**, 3 (2002), 543–556.

[28] P. W. Holland and S. Leinhardt
An exponential family of probability distributions for directed graphs (with discussion).
Journal of the American Statistical Association, **76**, 373 (1981), 33–50.

[29] J. Kittler, W. J. Christmas, and M. Petrou
Probabilistic relaxation for matching problems in machine vision.
Proceedings of Fourth International Conference on Computer Vision, 1993, pp. 666–673.

[30] D. Koller and A. Pfeffer
Probabilistic frame-based systems.
Proceedings of the Fifteenth National Conference on Artificial Intelligence, 1998, pp. 580–587.

[31] G. Levchuk, C. Chopra, and K. Pattipati
NetSTAR: Methodology to identify enemy network structure, tasks, activities, and roles.
10th International Command and Control Research and Technology Symposium, McLean, VA, June 13–16, 2005.

[32] Y. Li, P. Pardalos, and M. Resende
A greedy randomized adaptive search procedure for the quadratic assignment problem.
The Center for Discrete Mathematics and Theoretical Computer Science Series in Discrete Mathematics and Theoretical Computer Science, 1994.

[33] E. Loiola, N. Abreu, P. Boaventura-Netto, P. Hahn, and T. Querido
A survey for the quadratic assignment problem.
European Journal of Operational Research, **176** (2007), 657–690.

[34] B. Luo and E. R. Hanckock
Structural graph matching using the EM algorithm and singular value decomposition.
IEEE Transactions on Pattern Analyses and Machine Intelligence, **23**, 10 (2001), 1120–1136.

[35] T. Mavridou, P. Pardalos, L. Pitsoulis, and M. Resende
A GRASP for the bi-quadratic assignment problem.
Theory and Methodology, **105** (1998), 613–621.

[36] A. Misevicius
A modified simulated annealing algorithm for the quadratic assignment problem.
Informatica, 2003.

[37] R. J. Mooney, P. Melville, L. R. Tang, J. Shavlik, I. D. Castro Dutra, D. Page, and V. S. Costa
Relational data mining with inductive logic programming for link discovery.
In *Proceedings of the National Science Foundation Workshop on Next Generation Data Mining*, Baltimore, MD, Nov. 2002.

[38] H. Oinas-Kukkonen, K. Lyytinen, and Y. Yoo
Social networks and information systems: Ongoing and future research streams.
Journal of the Association for Information Systems, **11** (Feb. 2010), special issue, 61–68.

[39] K. Pattipati, P. Willett, J. Allanach, H. Tu, and S. Singh
Hidden Markov models and bayesian networks for counter-terrorism.
In *R. Popp, J.Yen. Emergent Information Technologies and Enabling Policies for Counter-Terrorism*, Wiley-IEEE Press, June 2006.

[40] A. Perchant, C. Boeres, I. Bloch, M. Roux, and C. Ribeiro
Model-based scene recognition using graph fuzzy homomorphism solved by genetic algorithm.
2nd International Workshop on Graph-Based Representations in Pattern Recognition, Castle of Haindorf, Austria, 1999, pp. 61–70.

[41] R. Popp, K. Pattipati, and Y. Bar-Shalom
An m-Best S-D assignment algorithm and parallelization with application to multitarget tracking.
IEEE Transactions on Aerospace and Electronic Systems, **37** (Jan. 2001), 22–39.

[42] M. Resends, K. Ramakrishnan, and Z. Drezner
Computing lower bounds for the quadratic assignment problem with an interior point algorithm for linear programming.
Operations Research, **43** (1995), 781–791.

[43] G. L. Scott and H. C. Longuet-Higgins
An algorithm for associating the features of 2 images.
In *Proceedings of Royal Society, London Series B*, **244**, 1309 (1991), 21–26.

[44] L. S. Shapiro and J. M. Brady
Feature-based correspondence—An eigenvector approach.
Image and Vision Computing, **10** (1992), 283–288.

[45] D. Skillicorn
Social network analysis via matrix decompositions: Al Qaeda.
Available from http://www.cs.queensu.ca/home/skill/alqaeda.pdf, Aug. 2004.

[46] H. Tu, J. Allanach, S. Singh, P. Willett, and K. Pattipati
Information integration via hierarchical and hybrid Bayesian networks.
IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans, special issue on Advances in Heterogeneous and Complex System Integration, **1**, 1 (Jan. 2006), 19–34.

[47] T. Urban
Solution procedures for the dynamic facility layout problem.
Annals of Operations Research, **76** (1998), 323–342.

[48] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky
MAP estimation via Agreement on (hyper) trees: Message-passing and linear-programming approaches.
IEEE transactions on Information Theory, **51**, 11 (Oct. 2005), 3697–3717.

[49] S. Wasserman
Logit models and logistic regressions for social networks: I. An introduction to Markov random graphs and p^*.
Psychometrika, **60** (1996), 3–18.

[50] R. C. Wilson and E. R. Hancock
A Bayesian compatibility model for graph matching.
Pattern Recognition Letters, **17** (1996), 263–276.

[51] Y. Xia and Y. Yuan
A new linearization method for quadratic assignment problems.
Optimization Methods and Software, **21** (2006), 805–818.

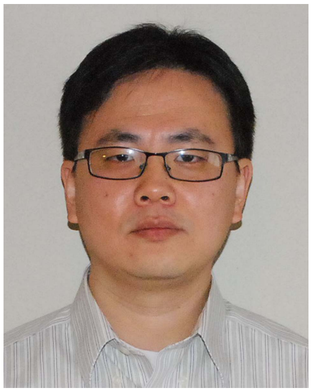
[52] J. J. Xu and H. Chen
CrimeNet explorer: A framework for criminal network knowledge discovery.
Association for Computing Machinery Transactions on Information Systems, **23**, 2 (Apr. 2005), 201–226.

[53] F. Yu, G. Levchuk, F. Tu, and K. R. Pattipati
A probabilistic computational model for identifying organizational structures from uncertain message data.
10th international Conference on Information Fusion, Quebec, Canada, 2007.

[54] Y. Zhang, M. Brady, and S. Smith
Segmentation of brain MR images through a hidden Markov random field model and the expectation maximization algorithm.
IEEE Transactions on Medical Imaging, **20**, 1 (2001), 45–57.

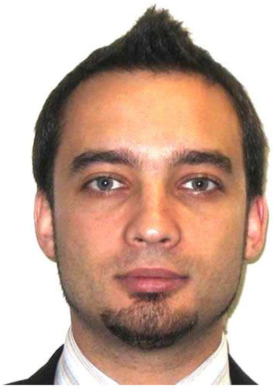
Xu Han received the B.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2005 and the M.S. degree in electrical engineering from the University of Connecticut, Storrs, in 2009, where she is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering.

Her research interests include the application of optimization techniques and algorithms to agent-based distributed planning in multi-echelon command and control systems, network identification and tracking, and adaptive organizations.



Feili Yu (M'03) received the B.S. degree in optical instrument engineering and the M.S. degree in information & electronic systems from Zhejiang University, Hangzhou, China, in 1994 and 1999, respectively, and the Ph.D. degree from the University of Connecticut in 2007.

Dr. Yu is currently with the Science Application International Corporation (SAIC) as a senior research engineer. Prior to SAIC, he was a system analyst in the independent system operation of New England, Inc. (ISO New England). His primary research interests are system modeling and simulation using optimization technologies, multi-model forecasting, and knowledge discovery from meta-data.



Georgiy M. Levchuk received the B.S./M.S.M. in mathematics (with highest honors) from the National Taras Shevchenko University of Kiev, Kiev, Ukraine, in 1995 and the Ph.D. degree in electrical engineering from the University of Connecticut, Storrs, in 2003.

He is currently a distinguished engineer in Analytics, Modeling, and Simulation Division at Aptima, Inc., Woburn, MA. Prior to joining Aptima, he held a research assistant position at the Institute of Mathematics, Kiev; a teaching assistantship at Northeastern University, Boston, MA; and a research assistantship at the University of Connecticut. His research interests include statistical classifiers, machine learning and probabilistic pattern recognition models to support information extraction, reasoning, and decision making in highly uncertain and ambiguous environments. He is developed algorithms to learn and identify multi-attributed networks from multi-source large-scale datasets, including finding hostile resource networks and critical enemy actors, mapping the activity networks to geo-spatial data under high uncertainty conditions, exploiting persistent video surveillance to recognize complex spatiotemporal activities and functions of interdependent moving and static entities, and using probabilistic network-based text mining to find entity-relation patterns in text. Dr. Levchuk was the recipient of the Best Student Paper Awards at the 2002 and 2003 Command and Control Research and Technology Symposia and the Best Paper Award at the 2004 Command and Control Research and Technology Symposium.

Krishna R. Pattipati (S'77—M'80—SM'91—F'95) received the B.Tech. degree in electrical engineering with highest honors from the Indian Institute of Technology, Kharagpur, in 1975, and the M.S. and Ph.D. degrees in systems engineering from the University of Connecticut in 1977 and 1980, respectively.

From 1980–1986 he was employed by ALPHATECH, Inc., Burlington, MA. Since 1986, he has been with the University of Connecticut, where he is currently the UTC Professor of Systems Engineering in the Department of Electrical and Computer Engineering. He has served as a consultant to Alphatech, Inc., Aptima, Inc., and IBM Research and Development. He is a cofounder of Qualtech Systems, Inc., a small business specializing in intelligent diagnostic software tools. His current research interests are in the areas of agile planning in dynamic and uncertain environments, diagnosis and prognosis techniques for complex system monitoring, and predictive analytics for threat detection. He has published over 400 articles, primarily in the application of systems theory and optimization (continuous and discrete) techniques to large-scale systems.

Dr. Pattipati was selected by the IEEE Systems, Man, and Cybernetics Society as the Outstanding Young Engineer of 1984, and received the Centennial Key to the Future award. He has served as the Editor-in-Chief of the *IEEE Transactions on SMC: Part B—Cybernetics* during 1998–2001, vice-president for Technical Activities of the IEEE SMC Society (1998–1999), and as vice-president for Conferences and Meetings of the IEEE SMC Society (2000–2001). He was corecipient of the Andrew P. Sage award for the Best SMC Transactions Paper for 1999, Barry Carlton award for the Best AES Transactions Paper for 2000, the 2002 and 2008 NASA Space Act Awards for "A Comprehensive Toolset for Model-based Health Monitoring and Diagnosis," the 2003 AAUP Research Excellence Award and the 2005 School of Engineering Teaching Excellence Award at the University of Connecticut. He also won the best technical paper awards at the 1985, 1990, 1994, 2002, 2004, 2005, and 2011 IEEE AUTOTEST Conferences, and at the 1997 and 2004 Command and Control Conferences.

Fang Tu (M'02) received the B.S. degree in optical engineering and the M.S. degree in information and communication electronics from Zhejiang University, Hangzhou, China, in 1994 and 1998, respectively, and the Ph.D. degree in electrical engineering from University of Connecticut, Storrs, CT, in 2003.

She is currently with the Global Service Technology in GE Healthcare, WI. Her research interests include data mining, optimization, statistical analysis, and signal processing with particular emphasis on anomaly detection and predictive analytics.

Corrections to "A Critical Look at the PMHT"

DAVID F. CROUSE
LE ZHENG
PETER WILLETT

In the above paper [1], a constant was missing from equation (100) for $\pi_m(n)$, the simplified form of equation (97) for the prior association probabilities involving hypergeometric functions. Without this constant, the resulting probabilities can be invalid, i.e., negative. This mistake carried through to equation (19) in the body of the text and into the formulation of $\bar{\pi}$, the ratio of π_0 to $\pi_{m\neq 0}$ in equation (102). It also led to incorrect entries results in Table II.

Equation (97) of [1] is

$$\pi_m(n)\big|_{m\neq 0} = \frac{\sum_{k=1}^{\min(n,M)} \dfrac{k}{(\lambda V)^k (n-k)!}\dbinom{M}{k} P_D^k (1-P_D)^{M-k}}{Mn\sum_{i=0}^{\min(n,M)} \dfrac{1}{(\lambda V)^i (n-i)!}\dbinom{M}{i} P_D^i (1-P_D)^{M-i}}. \tag{1}$$

The instructions for rewriting the types of sums given in the numerator and denominator are given in [2]. The correct form of equations (19) and (100) in [1] is thus

$$\pi_m(n) = \begin{cases} 1 - M\pi_1(n) & m = 0 \\ \left(\dfrac{P_D}{(1-P_D)\lambda V}\right) \dfrac{{}_2F_0\left[1-M, 1-n; \dfrac{P_D}{(1-P_D)\lambda V}\right]}{{}_2F_0\left[-M, -n; \dfrac{P_D}{(1-P_D)\lambda V}\right]} & \\ & m \neq 0 \end{cases}. \tag{2}$$

The ratio $\pi_0/\pi_{m\neq 0}$ in equation (102) should consequently be

$$\bar{\pi} = -M + \left(\frac{(1-P_D)\lambda V}{P_D}\right) \times \frac{{}_2F_0\left[-M, -n; \dfrac{P_D}{(1-P_D)\lambda V}\right]}{{}_2F_0\left[1-M, 1-n; \dfrac{P_D}{(1-P_D)\lambda V}\right]} \tag{3}$$

and the corrected form of Table II is given in Table I of this note

TABLE I
Corrected Form of Table II from [1]

M	$\pi_m(n)\big\|_{m\neq 0}$
1	$\dfrac{P_D}{nP_D + (1-P_D)\lambda V}$
2	$\dfrac{P_D^2(n-1-\lambda V) + P_D\lambda V}{P_D^2 n(n-1) - 2nP_D(P_D-1)\lambda V + (\lambda V)^2(P_D-1)^2}$
M	$\bar{\pi}$
1	$\left(\dfrac{1}{P_D} - 1\right)\lambda V + n - 1$
2	$\dfrac{(\lambda V)^2(1-P_D)^2 - 2\lambda V(n-1)(P_D-1)P_D + (n-2)(n-1)P_D^2}{P_D\lambda V - P_D^2(1+\lambda V - n)}$

Manuscript received July 21, 2011; released for publication August 10, 2011.

Refereeing of this contribution was handled Dr. Stefano Coraluppi.

Authors' addresses: D. F. Crouse and P. Willett, Department of Electrical and Computer Engineering, University of Connecticut, 371 Fairfield Way, U-2157, Storrs, CT 06269, (E-mail: {crouse, willett}@engr.uconn.edu); L. Zheng, School of Information and Electronics at the Beijing Institute of Technology, China, (E-mail: lezheng8451@163.com).

1557-6418/12/$17.00 © 2012 JAIF

REFERENCES

[1] D. F. Crouse, M. Guerriero, and P. Willett
A critical look at the PMHT.
Journal of Advances in Information Fusion, **4**, 2 (Dec. 2009), 93–116.

[2] M. Petkovšek, H. S. Wilf, and D. Zeilberger
$A = B$.
Wellesley, MA: A K Peters, Ltd., 1996, ch. 3.3. [Online], available: http://www.math.upenn.edu/wilf/AeqB.html.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

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

2012 BOARD OF DIRECTORS*

2010–2012	2011–2013	2012–2014
Simon Maskell	Sten F. Andler	Darin T. Dunham
Peter Willett	Yvo Boers	Fredrik Gustafsson
Wolfgang Koch	Lyudmila Mihaylova	Lance M. Kaplan

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

PAST PRESIDENTS

Joachim Biermann, 2011
Stefano Coraluppi, 2010
Elisa Shahbazian, 2009
Darko Musicki, 2008
Erik Blasch, 2007
Pierre Valin, 2006
W. Dale Blair, 2005
Chee Chong, 2004
Xiao-Rong Li, 2003
Yaakov Bar-Shalom, 2002
Pramod Varshney, 2001
Yaakov Bar-Shalom, 2000
Jim Llinas, 1999
Jim Llinas, 1998

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 7 Number 1 June 2012 ISSN 1557-6418