December 2011 Volume 6 Number 2 ISSN 1557-6418

JAIF

Journal of Advances in Information Fusion

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

Regular Papers Page

Track-to-Track Association and Ambiguity Management in the Presence of Sensor Bias 77
Dimitri Papageorgiou, Georgia Institute of Technology
Michael Holender, The Raytheon Company

Decision-Level Fusion Performance Improvement From Enhanced HRR Radar Clutter Suppression 101
Bart Kahler, United States Air Force Research Laboratory (AFRL)
Erik Blasch, United States Air Force Research Laboratory (AFRL)

Estimating Network Parameters for Selecting Community Detection Algorithms 119
Leto Peel, BAE Systems Advanced Technology Centre, UK

Heterogeneous Track-to-Track Fusion 131
Ting Yuan, University of Connecticut, USA
Yaakov Bar-Shalom, University of Connecticut, USA
Xin Tian, University of Connecticut, USA

Characterization and Empirical Evaluation of Bayesian and Credal Combination Operators 150
Alexander Karlsson, University of Skövde, Sweden
Ronnie Johansson, University of Skövde, Sweden
Sten F. Andler, University of Skövde, Sweden

Detecting a Small Boat using Histogram PMHT 167
Samuel Davey, Defence Science and Technology Organisation, Australia

Real-time Allocation of Firing Units To Hostile Targets 187
Fredrik Johansson, Swedish Defence Research Agency, Sweden
Göran Falkman, University of Skövde, Sweden

From the Editor-In-Chief

Where Are the Design Methods for Information Fusion Algorithms?

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 (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

December 2011

Where Are the Design Methods for Information Fusion Algorithms?

The community of information fusion researchers is a very prolific group that publishes papers in numerous annual conferences and journals. However, the number of success stories associated with fielding systems that include information fusion is relatively small. When considering the low rate of transitions of information fusion techniques to real-world systems, I noticed that for all practical purposes no design methods exist 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.

According to the Accreditation Board for Engineering and Technology (ABET), engineering design is the process of devising a system, component, or process to meet desired needs. It is a decision-making process (often iterative), in which the basic sciences, mathematics, and the engineering sciences are applied to convert resources optimally to meet these stated needs. This ABET guidance implies a design process that achieves a guaranteed performance. The technical domain of information fusion includes very little for which engineers can draw upon in their design process and almost none of the information fusion algorithms have any performance guarantees for which the designer can utilize to ensure the needs of the design will be met the needs or requirements. In this editorial, I focus on the need for design processes, while system requirements and performance guarantees will be discussed in the June 2012 editorial.

Until recently [1, 2], the wealth of literature on the well-studied and rather straightforward problem of tracking maneuvering targets did not include design methods. The model mismatch present in the Kalman filter (zero-mean white process noise) to model (deterministic but unknown) target maneuvers complicates the design process, and the lack of attention from the information fusion community on a design method has

left this as an open problem for over 30 years. Prior to [1, 2], conventional wisdom for the design of nearly-constant filters for tracking maneuvering targets [3] suggests that the process noise standard deviation should be chosen greater than one half the maximum acceleration of the target and less than the maximum acceleration. This guidance was based on experience and the design criterion was never quantified. In fact, using this design guidance can lead one to design a track filter that produces errors greater than the errors in the sensor measurements [4]. In [1, 2], the maximum acceleration of the target and duration of the maneuvers along with the sensor parameters were used to define upper and lower limits on the process noise standard deviation. The lower limit on the process noise standard deviation is defined to prevent the estimation errors from exceeding the measurement errors, while the upper limit is set to minimize the maximum mean squared error. Thus, a systematic method for the design of nearly constant velocity filters for tracking maneuvering targets is introduced in [1, 2]. That technique is extended to radar tracking in [5].

This problem of tracking maneuvering targets has more open issues with respect to design methods. For example, the choice between a nearly constant acceleration filter and a nearly constant velocity filter for a specific tracking problem is an open issue. The choice between a multiple model estimator and a Kalman filter is another open issue that was originally addressed in [6]. A shortcoming of the results in [6] is the need to select the process noise standard deviation which we know was an open issue during its publication. The technical domain of information fusion is full of open problems related to the design of information fusion algorithms. Some examples include selection of the gating threshold, choice between probabilistic data association and multiple hypothesis tracking (MHT), selection of the memory depth of an MHT, and selection of the costs of missed detections and no assignment in measurement-to-track assignment.

Development of effective design methods for information fusion algorithms is the next frontier for the information fusion community. Every implementation of information fusion cannot be treated as a new research problem, if information fusion algorithms are going to be implemented in real-world systems. Readily available design methods with performance guarantees are a prerequisite for the transition of information fusion methods into real-world systems. Development of design methods needs to become the newest research topic for the information fusion community.

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] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation with applications to tracking and navigation.
New York: John Wiley and Sons, Inc., 2001.

[4] 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), 450–456.

[5] 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.

[6] T. Kirubarajan and Y. Bar-Shalom
Kalman filter versus IMM estimator: When do we need the later.
IEEE Transactions on Aerospace and Electronic Systems, **39**, 4 (Oct. 2003), 1452–1457.

Track-to-Track Association and Ambiguity Management in the Presence of Sensor Bias

DIMITRI J. PAPAGEORGIOU
MICHAEL HOLENDER

The track-to-track association problem is to determine the pairing of sensor-level tracks that correspond to the same true target from which the sensor-level tracks originated. This problem is crucial for multisensor data fusion and is complicated by the presence of individual sensor biases, random errors, false tracks, and missed tracks. A popular approach to performing track-to-track association between two sensor systems is to jointly optimize the *a posteriori* relative bias estimate between the sensors and the likelihood of track-to-track association. Algorithms that solve this problem typically generate the K best bias-association hypotheses and corresponding bias-association likelihoods. In this paper, we extend the above approach in two ways. First, we derive a closed-form expression for computing "pure" track-to-track association likelihoods, as opposed to bias-association likelihoods which are weighted by a unique relative bias estimate. Second, we present an alternative formulation of the track-to-track association problem in which we optimize solely with respect to marginal association likelihoods. Finally, we provide two algorithms that find the K provably best track-to-track associations with respect to our new likelihood function. These results facilitate what is commonly known as system-level track ambiguity management.

Manuscript received June 25, 2010; revised January 25, 2011; released for publication April 26, 2011.

Refereeing of this contribution was handled by Dr. Stefano Coraluppi.

The first author was partially supported by a National Science Foundation Graduate Research Fellowship.

Authors addresses: D. J. Papageorgiou, H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, 765 Ferst Drive NW, Atlanta, GA 30332, E-mail: (djpapag@gatech.edu); M. Holender, The Raytheon Company, Integrated Defense Systems, 225 Presidential Way, Woburn, MA 01801, E-mail: (Michael_N_Holender@raytheon.com).

1. INTRODUCTION

A fundamental problem in multisensor data fusion is associating data from different sensor systems in the presence of sensor bias, random errors, false alarms, and missed detections. Level 1 of the JDL Fusion [10] model encompasses many different processes while attempting the overall goal of Object Refinement. Raw data are input into the system and the first problem that is considered is determining what that data refer to and which elements of data associate to one another. In our case, associating data is crucial to providing a coherent, integrated picture to the user, as well as pertinent track and attribute information about various targets of interest.

A typical multiple target tracking (MTT) system is composed of a suite of heterogeneous sensor systems and one or more fusion nodes which receive and process the raw data provided to the system. Each sensor processes its own data to generate and maintain sensor-level (also known as locally-fused) tracks, while each fusion node fuses the sensor-level tracks into a set of system-level tracks. This architecture is hierarchical in nature and has become a widely accepted system design choice in many circles because of the lack of single points of failure and information processing bottlenecks [21].

It is within the context of this hierarchical MTT system that the track-to-track association problem becomes so crucial. Before continuing forward with the fusion process, a fusion node must first have high confidence that the track data (from different sensors) to be combined correspond to the same target. A host of factors can complicate this association. First, participating sensors operating under different phenomenology may track different subsets of the truth. Second, closely-spaced objects may be difficult to resolve due to individual sensor sensitivity. A third issue concerns error due to sensor biases, which result from misalignment of measurement axes and sensor location error often arise and are difficult to estimate. Proper estimation and removal of this error is important to making correct assignments [3, 19, 20].

As discussed in [3], [11], and [19], there are several potential sources of error that bias removal algorithms attempt to rectify. Moore and Blair [19] group these sources into three broad categories: sensor errors, sensor/platform position and heading errors, and transformation errors from one sensor to another. Sensor registration attempts to estimate errors associated with biases that are constant or changing very slowly with time so that they can be removed before filtering takes place. Sensor biases, which may arise in both range and angle dimensions, often account for the majority of the total error. Moreover, they are typically the most difficult to estimate and eliminate. While range error can consist of offset and scaling errors, the offset error is typically the culprit of the error that sensor registration seeks to remove. In this paper, we only consider the presence of

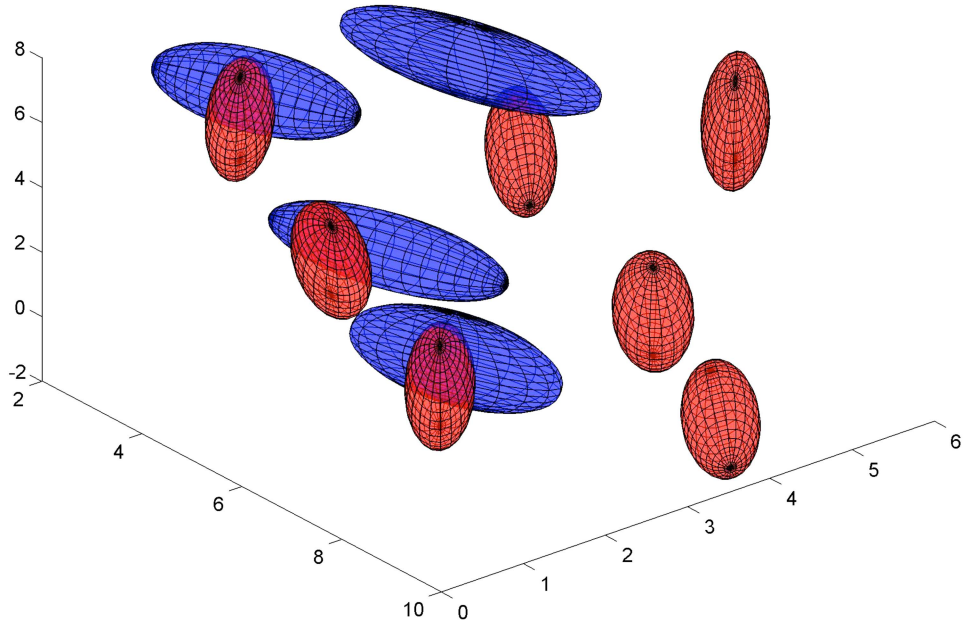


Fig. 1. The track-to-track association problem using positional densities only.

a random relative bias between sensors, without the aid of so-called *targets of opportunity* [15] or absolute bias estimates [12].

The track-to-track association problem is to determine the pairing of sensor-level tracks that correspond to the same true targets from which sensor-level tracks originated. The focus of this paper concerns track-to-track association between two sensor systems in the larger context of system-level tracking and discrimination. Fig. 1 depicts a simple example of the track-to-track association problem in which it is imperative to account for the presence of sensor bias. Sensor A sees four tracks, whereas sensor B sees seven, including the four seen by sensor A. The positional error volume of each sensor's tracks can be viewed as an ellipsoid in 3D. Despite the fact that several ellipsoids overlap, the true relative bias is such that no two corresponding track volumes overlap in the graphic.

Our point of departure is a track-to-track association problem posed by Levedahl [17], which he termed the Global Nearest Pattern Matching (GNPM) problem. The primary purpose of the GNPM problem was for a one-time sensor-to-sensor handover in which one sensor system transmits a frame of data to another sensor upon which a correlation algorithm is employed to correlate the information with its local track database. The novelty of Levedahl's approach, as well as Kenefic's [14] before him, was the explicit incorporation of a relative bias term into the likelihood function used to perform track-to-track association. For years, researchers and practitioners had acknowledged the existence of systematic sensor biases, but the vast majority of algorithms developed to resolve the problem were sequential in nature. That is, correlation algorithms would first attempt to estimate and remove the relative sensor bias, and then perform track-to-track association by solving a standard two-dimensional linear assignment problem (e.g., [27]). The fundamental drawback of this sequential approach lies in the infamous "chicken or the egg dilemma": The determination of the relative bias is inextricably linked with the correct track-to-track association and vice versa. Kenefic and Levedahl formally coupled the two problems and the resulting formulation has seen increasing use within the MTT community. A precise description of the GNPM problem is given in Section 2.

Algorithms that solve the GNPM problem typically generate the K best bias-association hypotheses and rank them according to their corresponding bias-association likelihood. One of the questions that we hope to address is: Is this information adequate for helping a system operator overseeing not just a suite of sensors, but weapon systems and other technology as well, make a more informed decision concerning whether or not to fuse the track data? In the air and missile defense domain, system-level tracking carries inherent uncertainties as heterogeneous sensors with different viewing geometries that observe closely-spaced objects disperse in a random manner. This inherent uncertainty in track formation and association has led to an area known as *track ambiguity management*.

Since identifying a single bias-association hypothesis whose likelihood "stands out" from the other hypotheses is an elusive task given this uncertainty, a secondary (and arguably less optimistic/more pragmatic) objective is to enumerate a subset of K bias-association hypotheses so that individual track pairings can be evaluated. Specifically, if there are a number of highly likely hypotheses whose likelihoods differ by negligible amounts, it is beneficial from a system-level per-

spective to know if there is a subset of track pairings that are common among all or most of the K best hypotheses. For example, in missile defense, one is often most interested in correctly associating a subset of the tracks, namely those that may represent objects of concern. Going a step further, it may be beneficial at the system-level for a correlation algorithm to quantify the likelihood that sensor A track i should be paired with sensor B track j. Unfortunately, one cannot produce these likelihoods using the bias-association likelihoods produced by an algorithm for the GNPM problem as will be discussed in Section 3.

In this paper, we argue that the information returned after solving the GNPM problem for track-to-track association may be insufficient in assisting a system operator make an informed decision concerning whether or not two tracks should be fused, especially in the case when a one-time sensor-to-sensor handover is necessary. Instead, we suggest that a closely-related, but different, likelihood function, which we call a *marginal track-to-track association likelihood function*, can be used either in lieu of the GNPM likelihood function or for the computation of individual track pairing likelihoods to improve system-level ambiguity management.

Our main contributions to the fusion community are the following: (1) We derive a closed-form solution for our proposed likelihood function and provide a procedure for computing likelihoods for individual track pairs. (2) We illustrate the usefulness of these track pair likelihoods with a detailed example based on four track scenes of increasing ambiguity. (3) We propose two exact algorithms for identifying the K provably best solutions to our new association problem, one of which may be suitable within an operational system.

The outline of this paper is as follows: In Section 2, a common set of assumptions is given and the GNPM problem is presented. In Section 3, we formally derive a closed-form expression for our marginal track-to-track association likelihood and contrast it with the GNPM likelihood function. A detailed example then follows of the practicality of the marginal track-to-track association likelihood function for system-level track ambiguity management. In Section 5, motivated by the findings presented in the example of Section 4, we propose an alternative formulation for performing track-to-track association using our marginal track-to-track association likelihood function in lieu of the GNPM likelihood function and two exact algorithms for solving this problem in Section 6. Computational results are presented in Section 7 followed by conclusions in Section 8.

2. PROBLEM DESCRIPTION

Consider two independent sensor systems—sensor A and sensor B—tracking an unknown number of targets in space. Let $N_A = \{1,\ldots,n_A\}$ and $N_B = \{1,\ldots,n_B\}$ denote the set of tracks formed by sensors A and B, respectively. Without loss of generality, we assume throughout that $n_A \leq n_B$. Due to a host of factors, including geometry and sensor resolution, the number of tracks formed by sensors A and B will often differ from the true number of targets and from one another. Let $\mathbf{x}_i^A$ and $\mathbf{P}_i$, for $i \in N_A$, denote the state estimate and error covariance matrix, respectively, of the ith sensor A track. Similarly, let $\mathbf{x}_j^B$ and $\mathbf{Q}_j$, for $j \in N_B$, denote the state estimate and covariance matrix, respectively, of the jth sensor B track. We assume that estimation errors for each sensor reporting on a common target are uncorrelated. This is only an approximation since, in general, track errors from different sensors are correlated. We assume all state estimates and covariance matrices have been extrapolated to a common time point and have been converted to a common D-dimensional reference frame.[1]

A key assumption in this track-to-track association framework is that each track state is corrupted by a constant, but unknown, sensor bias[2] [5, 17, 21, 25]. Ideally, these individual sensor biases would be estimated and removed prior to performing track-to-track association, but this is not always possible [3, 19]. Consequently, a distinguishing facet of this approach is our attempt to estimate the inter-sensor bias, or the relative bias, between the two sensors via maximum *a posteriori* (MAP) estimation. That is, this is a Bayesian estimation framework. The relative bias $\mathbf{b}$ is modeled as a Gaussian random vector having mean $\mathbf{0}$ and covariance $\mathbf{R}$ in a Cartesian coordinate frame. It is assumed that sensor bias only degrades a sensor's capability of measuring target state, and not its ability to detect a target.

We denote a track-to-track association by the vector $\mathbf{j}$. Consequently, the association of the ith track in N_A with the jth track in N_B is denoted by (i, j_i). It is convenient to think of the pair (i, j) as an undirected arc in a bipartite graph and the vector $\mathbf{j}$ as a compact notation for writing $\{(1, j_1),(2, j_2),\ldots,(n_A, j_{n_A})\}$. It is possible that the ith track in N_A is not assigned to any track in N_B, in which case we still write (i, j_i), but $j_i = 0$. We refer to such an assignment as a *null assignment*, or by saying that track i was assigned to the *dummy* track. We sometimes refer to partial and complete assignments. A *partial* assignment is one in which a strict subset of the sensor A tracks are assigned, while in a *complete* assignment, all sensor A tracks are assigned. A partial assignment can be made complete by assigning the currently unassigned sensor A tracks to the dummy track. It is implicitly assumed that at

[1] All vectors are column vectors. All vectors and matrices are written in bold font. All covariance matrices are assumed positive definite.

[2] It is worth noting that the assumption that average biases in synchronized data, transformed to a common reference frame, are equal is seldom true, even for identical sensors, unless they are collocated. This is the case for radars because during the process of spatial transformation to a common coordinate frame, the two biases get magnified nonlinearly and differently, as the latitudes, longitudes, and the ECR coordinates of the two sensors are different. Hence, this assumption is an approximation.

most one sensor A track can be assigned to a sensor B track and vice versa. We refer to the pair $(\mathbf{b},\mathbf{j})$ as a bias-association hypothesis, a hypothesis, or a solution to the GNPM problem.

A popular objective for track-to-track association is to simultaneously find the most likely track-to-track association and relative bias estimate. To do so, a likelihood function is needed to compare different solutions. Here, we follow the derivation of the likelihood function given in [17]. A general derivation of the *a posteriori* joint-probability-mass-density mixture function for more than two sensors is given in [16] before being specialized to the case of two sensors. The likelihood function for the GNPM problem is based upon the marriage of an *a posteriori* bias estimation problem and the standard two sensor track-to-track association problem. The first term

$$\frac{e^{-\mathbf{b}^T\mathbf{R}^{-1}\mathbf{b}/2}}{(2\pi)^{D/2}\sqrt{|\mathbf{R}|}}$$

(where $|\mathbf{R}|$ denotes the determinant of $\mathbf{R}$) is nothing more than a prior probability density on the relative bias, which we assume is available. We refer the reader to [5] for further discussion on the construction of priors in this framework. The second term consists of the product of the incremental likelihoods of track assignment. Specifically, given a bias estimate $\mathbf{b}$, the likelihood of assigning track $i \in N_A$ and track $j \in N_B$ is

$$\beta_T P_{AB} \frac{e^{-d_{ij}^2(\mathbf{b})/2}}{(2\pi)^{D/2}\sqrt{|\mathbf{S}_{ij}|}}$$

where

- β_T is the target density, i.e., the number of targets per unit volume in D-dimensional space;
- P_{AB} is the probability that a target is tracked by sensor A and sensor B;
- $\mathbf{S}_{ij} = \mathbf{P}_i + \mathbf{Q}_j$;
- $d_{ij}^2(\mathbf{b}) = (\mathbf{x}_i^A - \mathbf{x}_j^B - \mathbf{b})^T \mathbf{S}_{ij}^{-1}(\mathbf{x}_i^A - \mathbf{x}_j^B - \mathbf{b})$ is the squared Mahalanobis distance between tracks i and j, parameterized by a bias estimate $\mathbf{b}$.

It is also possible for track $i \in N_A$ to be unassigned, in which case the incremental likelihood is the null assignment likelihood $\beta_{NTA}\beta_{NTB}$, where

- $\beta_{NTA} = \beta_T P_{AB}^{-} + \beta_{FA}$ represents a target density of no target existing for sensor A, and P_{AB}^{-} is the probability of tracking an object with sensor B but not with sensor A;
- $\beta_{NTB} = \beta_T P_{A\bar{B}} + \beta_{FB}$ represents a target density of no target existing for sensor B, and $P_{A\bar{B}}$ is the probability of tracking an object with sensor A but not with sensor B;
- the densities β_{FA} and β_{FB} represent the false track densities for sensor A and B, respectively. False tracks are not uncommon when tracking extended objects, i.e., objects for which a sensor may receive multiple detections on a given data frame.

Multiplying these likelihoods together, we arrive at the *GNPM likelihood function*:

$$L(\mathbf{b},\mathbf{j}) = \frac{e^{-\mathbf{b}^T\mathbf{R}^{-1}\mathbf{b}/2}}{(2\pi)^{D/2}\sqrt{|\mathbf{R}|}} \times \prod_{i=1}^{n_A} \left\{ \begin{array}{ll} \beta_T P_{AB} \dfrac{e^{-d_{ij_i}^2(\mathbf{b})/2}}{(2\pi)^{D/2}\sqrt{|\mathbf{S}_{ij_i}|}} & \text{if} \quad j_i > 0 \\ \beta_{NTA}\beta_{NTB} & \text{if} \quad j_i = 0 \end{array} \right\}. \tag{1}$$

It should be noted that we, like many others, abuse terminology by referring to $L(\mathbf{b},\mathbf{j})$ as a likelihood function, when, in fact, it is a posterior joint-probability-mass-density mixture function as in Corollary 1 in [16], although our $L(\mathbf{b},\mathbf{j})$ differs by a factor from that given in Corollary 1 in [16]. Note that we have assumed that assignment likelihoods for track pairs are independent. From a computational perspective, it is more convenient to work with the negative log likelihood. After some algebra and the removal of unnecessary constants, we obtain a modified version of the negative log likelihood function

$$-\log L(\mathbf{b},\mathbf{j}) = \mathbf{b}^T\mathbf{R}^{-1}\mathbf{b} + \sum_{i=1}^{n_A} c_{ij_i}(\mathbf{b}) \tag{2}$$

where

$$c_{ij}(\mathbf{b}) = \begin{cases} d_{ij}^2(\mathbf{b}) + \log|\mathbf{S}_{ij}| & \text{if} \quad j \in N_B \\ g & \text{if} \quad j = 0 \end{cases}$$

and

$$g = -2\log\left(\frac{\beta_{NTA}\beta_{NTB}(2\pi)^{D/2}}{\beta_T P_{AB}}\right) \tag{3}$$

is the so-called (log likelihood) *gate value*, which can be interpreted as a cost incurred for assigning the ith sensor A track to the dummy track $j = 0$. Extensions for feature-aided association have been made (see, e.g., [2, 7, 27]), but this topic lies beyond the scope of this work.

Having characterized the fundamentals of how a bias-association hypothesis is "scored" via the GNPM likelihood function, we now explicitly formulate the GNPM problem as a mathematical program, specifically as a mixed-integer nonlinear program (MINLP) [5, 25]. This formulation will be necessary in Section 5 as we contrast it with an alternative formulation. The standard assumption in track-to-track association, which we follow, is that each sensor A track can be assigned to at most one sensor B track and vice versa. After introducing binary decision variables y_{ij}, for $i = 1,\ldots,n_A$ and for $j = 0,\ldots,n_B$, such that y_{ij} takes value one if sensor A track i is assigned to sensor B track j (or possibly the dummy track $j = 0$), and is zero otherwise, we can now cast the *GNPM problem* as the following constrained

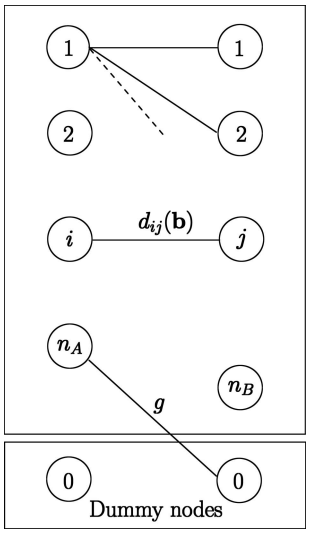


Fig. 2. Undirected bipartite graph.

minimization problem:

$$\begin{aligned} \min \quad & \mathbf{b}^T\mathbf{R}^{-1}\mathbf{b} + \sum_{i=1}^{n_A}\sum_{j=0}^{n_B} c_{ij}(\mathbf{b})y_{ij} \\ \text{s.t.} \quad & \mathbf{y} \in \mathcal{Y}, \quad \mathbf{b} \in \mathbb{R}^D \end{aligned} \tag{4}$$

where $\mathcal{Y} = \{\mathbf{y} \in \{0,1\}^{n_A \times (n_B+1)} : \sum_{j=0}^{n_B} y_{ij} = 1, \text{ for } i \in N_A, \sum_{i=1}^{n_A} y_{ij} \leq 1, \text{ for } j \in N_B\}$ is the set of all feasible track-to-track associations and $\mathbf{y}$ is a vectorized form of the y_{ij} variables. The above notation is standard in mathematical programming.

Basic characteristics and important observations of the GNPM problem are provided in [25]. Although Danford et al. [5] offer a similar formulation which they classify as a modified general network flow problem, we prefer to think of the GNPM problem as a two-dimensional nonlinear assignment problem, which is an extension of the traditional two-dimensional linear assignment problem used to perform track-to-track association. For those unfamiliar with mathematical programming, it may be convenient to interpret the GNPM problem (4) as a matching problem on a bipartite graph, depicted in Fig. 2, in which the objective is to minimize the sum of total arc costs subject to the constraint that each sensor A track can be assigned to at most one sensor B track and vice versa, where the arc costs are a function of the relative bias vector $\mathbf{b}$, and, hence, are nonlinear in the decision variables. Note that in Fig. 2 each non-dummy node on the left is connected to each non-dummy node on the right with a cost of $c_{ij}(\mathbf{b})$. Each dummy node is also connected to every node on the opposite side with a cost of g.

3. DERIVATION OF A MARGINAL TRACK-TO-TRACK ASSOCIATION LIKELIHOOD

In certain circumstances, we may wish to rank association hypotheses with respect to a "pure" track-to-track association likelihood, which we refer to as a *marginal track-to-track association likelihood* or MTTA likelihood for short. In particular, we continue to assume that the relative bias is a Gaussian random vector, but rather than optimize the *joint* bias-association likelihood, we optimize only the marginal likelihood of track-to-track association. In addition to reporting MTTA likelihoods to a system-user, another purpose of isolating the likelihood solely in terms of a track-to-track association is to facilitate the computation of pairwise association likelihoods for track ambiguity management, as will be explained below. After the initial release of this paper, we learned that Ferry [9] also advocates using MTTA likelihoods, which he calls *exact association probabilities*, and provides a comprehensive derivation of what follows for the more general setting involving more than two sensors and feature data.

To obtain our desired MTTA likelihood function, we remove the likelihood term for the relative bias in Equation (1) by integrating over all possible bias estimates. This yields $L(\mathbf{j}) =$

$$\int_{\mathbf{b}\in\mathbb{R}^D} \prod_{i=1}^{n_A} \left\{ \begin{array}{ll} \beta_T P_{AB} \dfrac{e^{-d_{ij_i}^2(\mathbf{b})/2}}{(2\pi)^{D/2}\sqrt{|\mathbf{S}_{ij_i}|}} & \text{if} \quad j_i > 0 \\ \beta_{NTA}\beta_{NTB} & \text{if} \quad j_i = 0 \end{array} \right\} \times \frac{e^{-\mathbf{b}^T\mathbf{R}^{-1}\mathbf{b}/2}}{(2\pi)^{D/2}\sqrt{|\mathbf{R}|}} d\mathbf{b}. \tag{5}$$

We would like to show that for a given complete assignment $\mathbf{j}$, Equation (5) has a convenient closed-form solution. For simplicity, assume that $\beta_T P_{AB} = 1$, that $j_i > 0$ for all i, and let $\mathbf{x}_{ij_i} = \mathbf{x}_i^A - \mathbf{x}_{j_i}^B$ and $\mathbf{S}_{ij_i} = \mathbf{P}_i + \mathbf{Q}_{j_i}$, for $i \in N_A$. Then, Equation (5) becomes $L(\mathbf{j}) =$

$$\int_{\mathbf{b}\in\mathbb{R}^D} \prod_{i=1}^{n_A} \frac{e^{-(1/2)(\mathbf{x}_{ij_i}-\mathbf{b})^T\mathbf{S}_{ij_i}^{-1}(\mathbf{x}_{ij_i}-\mathbf{b})}}{(2\pi)^{D/2}\sqrt{|\mathbf{S}_{ij_i}|}} \frac{e^{-\mathbf{b}^T\mathbf{R}^{-1}\mathbf{b}/2}}{(2\pi)^{D/2}\sqrt{|\mathbf{R}|}} d\mathbf{b}. \tag{6}$$

Defining $\mathbf{x}_{0j_0} = \mathbf{0}$ and $\mathbf{S}_{0j_0} = \mathbf{R}$, Equation (6) becomes

$$L(\mathbf{j}) = \int_{\mathbf{b}\in\mathbb{R}^D} \prod_{i=0}^{n_A} \frac{e^{-(1/2)(\mathbf{b}-\mathbf{x}_{ij_i})^T\mathbf{S}_{ij_i}^{-1}(\mathbf{b}-\mathbf{x}_{ij_i})}}{(2\pi)^{D/2}\sqrt{|\mathbf{S}_{ij_i}|}} d\mathbf{b}. \tag{7}$$

Notice that if we let X_i denote a multivariate Gaussian random variable with dimension D, mean $\mathbf{x}_{ij_i}$, and covariance $\mathbf{S}_{ij_i}$, for $i = 0, \ldots, n_A$, then Equation (7) is nothing more than the integral of the product of $n_A + 1$ independent Gaussian random variables

$$L(\mathbf{j}) = \int_{\mathbf{b}\in\mathbb{R}^D} \prod_{i=0}^{n_A} f_{X_i}(\mathbf{b}) d\mathbf{b}.$$

It can be shown that

$$L(\mathbf{j}) = \frac{\sqrt{(2\pi)^D|\mathbf{V}|}}{\sqrt{\prod_{i=0}^{n}(2\pi)^D|\mathbf{S}_{ij_i}|}} e^{-(1/2)\zeta}$$

where $\mathbf{V} = (\sum_{i=0}^{n_A}\mathbf{S}_{ij_i}^{-1})^{-1}$, $\zeta = (\sum_{i=0}^{n_A}\mathbf{x}_{ij_i}^T\mathbf{S}_{ij_i}^{-1}\mathbf{x}_{ij_i}) - \mathbf{u}^T\mathbf{V}\mathbf{u}$, and $\mathbf{u} = \sum_{i=0}^{n_A}\mathbf{S}_{ij_i}^{-1}\mathbf{x}_{ij_i}$ (see [23] or [9]).

Returning to the more general case in Equation (5), let I denote the set of sensor A tracks assigned to a sensor B track, i.e., $I = \{i \in N_A : j_i > 0\}$, and let I^0 denote the set of unassigned sensor A tracks, i.e., $I^0 = \{i \in N_A : j_i = 0\}$. Let $I^+ = I \cup \{0\}$. Then,

$$L(\mathbf{j}) = (\beta_{NTA}\beta_{NTB})^{|I^0|}(\beta_T P_{AB})^{|I|} \times \frac{\sqrt{(2\pi)^D|\mathbf{V}|}}{\sqrt{\prod_{i\in I^+}(2\pi)^D|\mathbf{S}_{ij_i}|}} e^{-(1/2)\zeta} \quad (8)$$

where $\mathbf{V} = (\sum_{i\in I^+}\mathbf{S}_{ij_i}^{-1})^{-1}$, $\zeta = (\sum_{i\in I^+}\mathbf{x}_{ij_i}^T\mathbf{S}_{ij_i}^{-1}\mathbf{x}_{ij_i}) - \mathbf{u}^T\mathbf{V}\mathbf{u}$, and $\mathbf{u} = \sum_{i\in I^+}\mathbf{S}_{ij_i}^{-1}\mathbf{x}_{ij_i}$. Note that in the remainder of the paper, when we refer to the *MTTA likelihood function* we typically have Equation (8) in mind since it is in closed-form, but one can also think of Equation (5) since this is the typical form of a marginal density function.

3.1. Implications for Track Ambiguity Management

Track-to-track association is one of many components in a multi-sensor MTT system. The decision logic used to create a single integrated air picture from multi-sensor data can vary widely from system to system, battle manager to battle manager. For this reason, modern association algorithms are being asked to return quantities besides the single most likely association to facilitate track ambiguity management, i.e., to assist the highest-level decision maker in managing track ambiguities. In this paper, we contend that the following information should often be included in the output returned by such an algorithm: (1) The K best GNPM bias-association solutions, (2) the K best MTTA solutions, and (3) a table of individual track pairing likelihoods or probabilities.

Having derived a closed-form expression for a marginal association likelihood over all possible relative bias values, we are now in a position to describe how one can generate a confusion matrix of individual track pairing likelihoods. As argued in the introduction, possessing individual track pairing likelihoods can be beneficial at the system-level where inherent uncertainties make it difficult to rank one association of track sets over another. Fundamentally, individual track pairing likelihoods provide a system-level tracking and discrimination architecture a quantifiable level of confidence that certain objects of interest should be paired together.

Let $\mathbf{j}^1,\ldots,\mathbf{j}^r$ denote all r possible track-to-track association vectors and $L(\mathbf{j}^1),\ldots,L(\mathbf{j}^r)$ the corresponding likelihoods as computed in Equation (8). Let T_k denote the set of track pairings in the kth best association hypothesis, for $k = 1,\ldots,r$. For all (i,j) pairs of tracks with $i = 0,1,\ldots,n_A$ and $j = 0,1,\ldots,n_B$, we can compute a pairwise likelihood

$$L_{ij} = \frac{1}{L_N}\sum_{\substack{k=1:\\(i,j)\in T_k}}^{r} L(\mathbf{j}^k)$$

where $L_N = \sum_{k=1}^{r} L(\mathbf{j}^k)$ is a normalizing constant. Together these pairwise likelihoods form what is often called a *confusion matrix*.

An obvious intractability in the above calculation is that the correct pairwise likelihood L_{ij} and the correct normalizing likelihood require the explicit enumeration of all r possible track-to-track association hypotheses, which grows factorially in the number of tracks n_A and n_B as noted in [17] and [25]. Since exhaustive enumeration is all but impossible except when n_A and n_B are sufficiently small, a heuristic approach is to compute approximate pairwise likelihoods

$$\hat{L}_{ij} = \frac{1}{\hat{L}_N}\sum_{\substack{k=1:\\(i,j)\in T_k}}^{\hat{r}} L(\mathbf{j}^k) \quad (9)$$

where $\hat{r}$ $(< r)$ is the number of association hypotheses that are returned by an algorithm for solving the track-to-track association problem in a limited amount of time and $\hat{L}_N = \sum_{k=1}^{\hat{r}} L(\mathbf{j}^k)$ is an approximate normalizing constant.

Three remarks are in order. First, track-to-track likelihoods L_{ij} could also be computed using techniques employed in the joint probabilistic data association (JPDA) method for calculating track-to-measurement probabilities (see, e.g., [28]). The computational complexity of these methods is exponential rather than factorial. We have suggested the heuristic above since it does not require any additional computation once the K best MTTA solutions have been found, which are useful to the battle manager in their own right. Second, those familiar with multiple hypothesis tracking (MHT) might recognize that a similar heuristic approach for computing approximate track hypothesis likelihoods is commonly used in MHT since one cannot enumerate all possible detection-to-track associations (see, e.g., Chp. 16 of [3]). Third, note that it would be incorrect to use the bias-association likelihoods $L(\mathbf{b},\mathbf{j})$ in computing the pairwise likelihoods L_{ij} since each $L(\mathbf{b},\mathbf{j})$ is weighted by a unique relative bias probability. Since the relative bias estimate differs from association to association, it does not make sense to compute pairwise likelihoods in this manner. Moreover, empirically we and others (see, e.g., Ferry [8]) have found that even approximating pairwise likelihoods in this manner can lead to unsatisfactory results.

3.2. Comparison of the MTTA and GNPM Likelihood Functions

The purpose of this subsection is to answer the following question: When might one prefer the MTTA likelihood function over the GNPM likelihood function? Below we provide three possible answers.

The first reason why one might favor the MTTA likelihood function over the GNPM likelihood function is a matter of interpretation. Recall that the fundamental objective of track-to-track association is to deter-

mine which sensor-level tracks correspond to the same true target from which the sensor-level tracks originated. Those who interpret this question to mean "What track-to-track association has the highest likelihood when weighted by the best relative bias estimate for that association?" are implicitly favoring the GNPM likelihood function in which the contribution of the best relative bias estimate for each association is used to compute the likelihood. Equivalently, one is implicitly placing a nontrivial emphasis on the relative bias estimation problem as they are on the association problem. On the other hand, those who interpret the fundamental question as "What track-to-track association has the highest likelihood over all possible relative bias values?" are implicitly placing a primary focus on the association problem and a secondary (minimal) focus on the relative bias estimation problem.

The second reason, which is closely related to the first, is to eschew overconfidence in any one particular association hypothesis. In certain instances, the difference between the GNPM likelihood of the best and next best bias-association hypotheses is large, even though the relative bias estimate for the best hypothesis is extremely unlikely. In some cases, this large difference can be misleading when deciding which tracks to fuse. This phenomenon can occur when the track-to-track association likelihood component of the joint likelihood function is large enough to offset the small likelihood of the corresponding relative bias estimate, for example, when many tracks with small track errors align nearly perfectly after an extremely unlikely relative bias is chosen. On the other hand, when all relative bias estimates are considered, the difference between the best and next best hypothesis is often less pronounced, and the ordering of the best hypotheses is often different, as will be shown in Section 4. In this case, a decision logic which uses association probabilities will perceive the bias-association ordering as being overly confident in the best bias-association hypothesis. See Ferry [8, 9] for related concerns.

The third reason why the MTTA likelihood function may be preferable is to facilitate the computation of individual track pairing likelihoods. Specifically, after computing the K best MTTA hypotheses, the calculation of approximate track pairing likelihoods requires virtually no additional computational effort. Since the goal of computing pairwise track-to-track association likelihoods implicitly emphasizes the association aspect, it seems natural to use the MTTA likelihood function. At the same time, it does not make mathematical/probabilistic sense to use bias-association likelihoods since each likelihood is weighted by a unique bias estimate that is different from hypothesis to hypothesis.

In summary, we are not suggesting that the marginal track-to-track association likelihood function presented in this paper is universally "better" than the GNPM likelihood function. However, we do advocate that it should be used to compute pairwise track-to-track association likelihoods and that it is a viable option when performing track-to-track association. Indeed, it may be more appropriate in certain circumstances in which one is emphasizing the association aspect of the fusion problem.

4. AN ILLUSTRATIVE EXAMPLE

The goal of this section is to present four track scenes illustrating (1) the differences between the GNPM and the MTTA likelihood functions, and (2) the benefits of the MTTA likelihood function in generating pairwise track-to-track association likelihoods for aiding system-level track ambiguity management. Specifically, we draw attention to the way in which pairwise association likelihoods can assist a system-user make an informed decision concerning whether or not to fuse two tracks.

Fig. 3 depicts four track-to-track association problem instances (track scenes) of increasing difficulty. Only three-dimensional positions are estimated and positional error baskets are shown. Our goal is to associate the four sensor A tracks (in blue) with the sensor B tracks (in red). Assume that the cost of a null assignment is relatively large, but not astronomical, so as to encourage actual track-to-track assignments. Scene 1 (upper left) is very easy as there are four sensor A tracks and four sensor B tracks. The correct association is identifiable by the eye. Scene 2 (upper right) has a nontrivial relative bias present; however, it is still easy to identify which tracks should be paired together. Scene 3 (lower left) has three tracks which can easily be associated, but there are two sensor B tracks in the upper right corner of this scene that could be paired with the fourth sensor A track in the same region of the figure. It is quite realistic that such a track scene can arise within certain domains of interest. For example, in ballistic missile tracking, a split track could occur in which sensor A collects detections during a given phase of the trajectory of a missile and forms four tracks. At some later point in time, sensor B begins observing the missile and forms five tracks because one of the objects splits into two after leaving sensor A's field of view. Scene 4 (lower right) is ambiguous. In one case, the four blue tracks could be paired with their neighboring red tracks (the ones touching the outer edge of the ellipsoids). In another case, the blue tracks 2, 3, and 4 could be paired with the red tracks 5, 6, and 7, respectively, towards the back right of the cube. We will see later that the former association is preferred.

Associated with each track scene are two tables. The first table lists the hypothesis number, the association vector $\mathbf{j}$, the negative log GNPM likelihood $-\log L(\mathbf{b},\mathbf{j})$ as computed in Equation (2), and the negative log MTTA likelihood $-\log L(\mathbf{j})$, as would be computed by taking the negative log of Equation (5), corresponding to the top twenty hypotheses that would be returned by an algorithm for the GNPM problem. Note that these hypotheses are ordered in nondecreasing order accord-

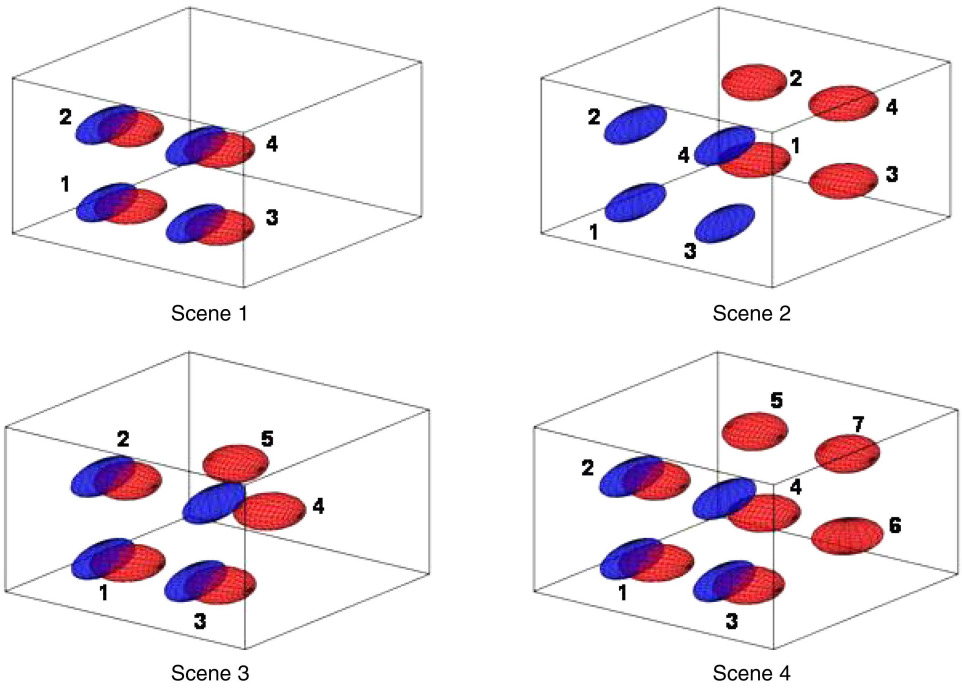


Fig. 3. Track scenes.

ing to $-\log L(\mathbf{b},\mathbf{j})$. As there are four track scenes, this information is presented in Tables I, III, V, and VII. The association vector is read as follows: Since there are four sensor A tracks in each scene, the assignment (j_1, j_2, j_3, j_4) means that sensor A track i is associated with sensor B track j_i, for $i = 1,\ldots,4$. If j_i is 0, then sensor A track i is unassigned (assigned to the dummy track). For example, the thirteenth best hypothesis (hypothesis 13) for track scene 1 has the assignment vector $(0,2,1,4)$, which means that sensor A track 1 is unassigned, while sensor A tracks 2, 3, and 4 are assigned to sensor B tracks 2, 1, and 4, respectively.

The second table associated with each track scene provides pairwise track-to-track association likelihoods as computed in Equation (9). Specifically, in Tables II, IV, VI, and VIII, one can look up in row i column j the likelihood that sensor A track i should be associated with sensor B track j. Similarly, the likelihood that sensor A track i should be unassociated (associated with a dummy track) is listed in row i and the column labeled 0. Note that in each row and column with a non-zero index the individual likelihoods sum to one. Although the pairwise likelihoods presented in these particular tables were computed using only the top twenty hypotheses, the pairwise likelihoods computed using all hypotheses gave the same result to six decimal places. As a matter of convenience, in the descriptions given for Tables I, III, V, and VII, the term "likelihood" should be interpreted to mean the negative log likelihood.

Track Scene 1: Very Easy

Table I provides the results for the K best ($K = 20$) hypotheses ordered with respect to the GNPM likelihood $-\log L(\mathbf{b},\mathbf{j})$. Notice that the first hypothesis has a significantly smaller negative log likelihood (hence, larger likelihood) than the next. This makes intuitive sense given that this particular scenario is very easy due to a small relative bias and a similar spatial configuration of the tracks from each sensor. Also, note that the ordering of the hypotheses would be slightly different if the MTTA likelihood $-\log L(\mathbf{j})$ were considered in lieu of the GNPM likelihood. This occurs throughout each of the four cases that we present and motivates the remaining sections of the paper.

Using the MTTA likelihoods $L(\mathbf{j})$, Table II was created to help the system-user understand the pairwise associations between tracks. As described in Section 3.1, one simply adds each $L(\mathbf{j})$ in the K-best hypotheses table for which the desired pairwise association exists and divide that value by the sum of all $L(\mathbf{j})$ values in the same table. In this example, the entries along the diagonal of Table II each achieve a high value of approximately 0.95 since the top hypothesis $\{(1,1),(2,2),(3,3),(4,4)\}$ is so much more likely than any other.

In considering a real-life fusion system, the end-user would find both of these tables useful in terms of assessing the current situation. The first table provides a list of complete track associations along with their relative likelihood values. This enables the user to consider all sensor tracks together. In addition, the user can utilize the second table to obtain information regarding individual track-to-track pairings. This can prove highly useful in a situation wherein the top two hypotheses are very close in terms of likelihood. In this case, it is certainly possible that the individual track-to-track associ-

TABLE I
Scene 1—Top 20 Bias-Association Hypotheses

Hyp	Assignment				$-\log L(\mathbf{b},\mathbf{j})$	$-\log L(\mathbf{j})$
1	1	2	3	4	15.6093	42.5388
2	1	0	3	4	25.0251	45.9716
3	1	2	3	0	25.0251	45.9717
4	0	2	3	4	25.0254	45.9717
5	1	2	0	4	25.0267	45.9722
6	2	1	3	4	28.1048	48.7875
7	1	2	4	3	28.1098	48.7890
8	1	4	3	2	28.1177	48.7936
9	3	2	1	4	28.1486	48.8081
10	1	0	3	2	29.3494	48.1344
11	2	0	3	4	29.3654	48.1421
12	1	2	4	0	29.3679	48.1430
13	0	2	1	4	29.3946	48.1560
14	0	1	3	4	29.8910	48.4052
15	3	2	0	4	29.8935	48.4055
16	1	2	0	3	29.8949	48.4064
17	1	4	3	0	29.9172	48.4179
18	1	0	2	4	33.6744	50.2973
19	1	0	4	2	33.6920	50.3055
20	2	0	1	4	33.7341	50.3264

TABLE II
Scene 1—Pairwise TTA Likelihood Table

	Sensor B				
Sensor A	1	2	3	4	0
1	0.9575	0.0052	0.0041	0.0000	0.0333
2	0.0041	0.9567	0.0000	0.0041	0.0352
3	0.0051	0.0004	0.9568	0.0052	0.0326
4	0.0000	0.0052	0.0041	0.9574	0.0333
0	0.0333	0.0326	0.0351	0.0334	0.0000

TABLE III
Scene 2—Top 20 Bias-Association Hypotheses

Hyp	Assignment				$-\log L(\mathbf{b},\mathbf{j})$	$-\log L(\mathbf{j})$
1	1	2	3	4	19.2874	44.2384
2	1	0	3	4	28.2314	47.4431
3	1	2	0	4	28.2361	47.4453
4	1	2	3	0	28.2388	47.4470
5	0	2	3	4	28.2438	47.4491
6	1	4	3	2	29.2532	49.2220
7	3	2	1	4	29.3967	49.2927
8	2	1	3	4	30.3936	49.7926
9	1	2	4	3	30.3991	49.7941
10	1	0	3	2	30.6948	48.6756
11	0	2	1	4	30.8173	48.7358
12	0	1	3	4	31.5877	49.1219
13	1	2	0	3	31.6002	49.1274
14	1	2	4	0	33.1373	49.8961
15	2	0	3	4	33.1447	49.9002
16	3	2	0	4	33.2380	49.9460
17	1	4	3	0	33.2405	49.9478
18	0	1	3	2	34.0486	50.3533
19	0	2	1	3	34.1829	50.4186
20	0	2	3	1	34.2054	50.4304

TABLE IV
Scene 2—Pairwise TTA Likelihood Table

	Sensor B				
Sensor A	1	2	3	4	0
1	0.9336	0.0059	0.0078	0.0000	0.0527
2	0.0110	0.9358	0.0000	0.0082	0.0450
3	0.0158	0.0000	0.9369	0.0059	0.0414
4	0.0016	0.0168	0.0109	0.9326	0.0381
0	0.0380	0.0415	0.0444	0.0533	0.0000

ation pairing table can shed some additional light onto the situation allowing the user to properly associate all tracks as needed.

Track Scene 2: Biased But Easy

Tables III and IV are configured in the same manner as Tables I and II. This will continue to be true for Scenes 3 and 4 in the following.

This scene has more bias than the previous and therefore the GNPM and MTTA likelihood values are slightly closer to one another with respect to the top two hypotheses. This logically provides values along the diagonal of Table IV that remain rather high, but are slightly lower than in the previous scene. This reduction in track pairing likelihoods along the main diagonal should be intuitively clear as the increased relative bias in the optimal hypotheses causes the contribution of the relative bias term in the likelihood functions to decrease (hence, the negative log likelihoods will increase). It is also worth noting that, just as in Track Scene 1, hypotheses 2 through 5 resemble the best hypothesis, hypothesis 1, except that only three of the four sensor A tracks are assigned, while the other is unassigned. If the gate value g were increased, these hypotheses, as well as all other hypotheses with null assignments, would become less favorable.

Track Scene 3: Slightly Ambiguous

This scene contains little bias but is more ambiguous than the previous two due to the existence of a fifth sensor B track. The results reflect this increased ambiguity in that the difference between hypotheses 1 and 2 is very slight. This is a case wherein the system-user may wish to look closely at Table VI for more information on the pairwise association confidence values. Although lower than in the previous two scenes, the diagonal values for tracks 1, 2 and 3 remain relatively high. However, the ambiguous portion (sensor A track 4 associates with either sensor B track 4 or 5) is readily observed as the values are 0.5043 and 0.4681, respectively. If the system-user is required to make a call regarding whether or not to fuse tracks, he may use this discrepancy to select $\{(1,1),(2,2),(3,3),(4,4)\}$ as the association vector. However, it is certainly acceptable in the fusion world to suggest that the results are inconclusive due to too much ambiguity. Indeed, in this scene, a reasonable alternative would be to fuse sensor A and B tracks 1, 2, and 3, while postponing

TABLE V
Scene 3—Top 20 Bias-Association Hypotheses

Hyp	Assignment				$-\log L(\mathbf{b},\mathbf{j})$	$-\log L(\mathbf{j})$
1	1	2	3	4	20.5577	41.3645
2	1	2	3	5	21.1202	41.6245
3	1	2	4	5	23.0836	42.6061
4	1	5	3	4	24.0485	43.0883
5	3	2	4	5	26.1490	44.1384
6	1	4	3	5	26.2722	44.2006
7	2	5	3	4	27.1338	44.6312
8	4	2	3	5	27.5214	44.8247
9	3	2	1	4	27.8584	45.0145
10	1	2	4	3	28.2426	45.2068
11	3	2	1	5	28.4286	45.2784
12	1	5	3	2	28.7029	45.4161
13	2	4	3	5	29.3567	45.7431
14	1	2	5	4	29.3879	45.7586
15	1	2	3	0	29.7613	44.7182
16	2	1	3	4	29.8116	45.9925
17	0	2	3	4	30.1844	44.9294
18	1	0	3	4	30.1853	44.9300
19	1	2	0	4	30.1870	44.9307
20	2	1	3	5	30.3742	46.2526

TABLE VI
Scene 3—Pairwise TTA Likelihood Table

	Sensor B					
Sensor A	1	2	3	4	5	0
1	0.9118	0.0254	0.0405	0.0117	0.0000	0.0106
2	0.0065	0.8689	0.0000	0.0266	0.0874	0.0106
3	0.0172	0.0000	0.8283	0.1393	0.0046	0.0106
4	0.0000	0.0065	0.0080	0.5043	0.4681	0.0131
0	0.0646	0.0992	0.1232	0.3180	0.4399	0.0000

a decision on sensor A track 4 and sensor B tracks 4 and 5.

Track Scene 4: Ambiguous

This scene is the most ambiguous of the four, which is reflected in the likelihoods of Tables VII and VIII. Tracks 1, 2, and 3 continue to have a high confidence of being the correct association. To some, the results may still look favorable, but to others the associations may not be strong enough to make a call. We leave these types of decision to the system-user.

With four concrete examples in hand, we close this section by briefly mentioning two ad hoc methods for computing individual track pairing likelihoods that were used in preliminary computational experiments [23]. We believe these approaches are inferior, but not altogether useless. First, it seems reasonable that one can get a general sense of how likely a particular track pairing is by simply counting the number of times sensor A track i is paired with sensor B track j among the top K hypotheses. For example, in the first track scene, sensor A track 1 is assigned to sensor B track 1 twelve times, to sensor B track 2 three times, to the dummy track three times, and to sensor B track 3 two times. Since sensor A track 1 and B track 1 are paired the most frequently, one might be willing to conclude that their data should be fused. Unfortunately, this approach quickly breaks down when considering the fourth track scene. In this scene, the most ambiguous of the four, we see that sensor A track 2 is assigned to sensor B tracks 2 and 5 ten times each. The "tallying" method is inadequate in this setting.

A second option is to augment the "tallying" method by giving each hypothesis an additional weight (as opposed to the uniform weight $1/K$ above). Rather than just count the number of times sensor A track i is paired with sensor B track j, one could choose

TABLE VII
Scene 4—Top 20 Bias-Association Hypotheses

Hyp	Assignment				$-\log L(\mathbf{b},\mathbf{j})$	$-\log L(\mathbf{j})$
1	1	2	3	4	19.6996	38.2521
2	1	2	3	7	21.5482	39.1769
3	1	2	6	4	21.7231	39.2636
4	1	5	3	4	21.8086	39.3065
5	1	2	3	5	21.9579	39.3816
6	1	2	3	6	22.3714	39.5884
7	1	5	6	7	22.4929	39.6492
8	1	2	6	7	22.5048	39.6553
9	1	5	3	7	22.5512	39.6784
10	1	5	6	4	22.8158	39.8099
11	1	2	4	7	23.2260	40.0157
12	1	5	4	7	23.4778	40.1416
13	1	2	6	5	23.5040	40.1546
14	1	2	4	5	23.6384	40.2217
15	2	5	6	7	23.9242	40.3653
16	2	5	4	7	24.1012	40.4535
17	3	5	6	7	24.1288	40.4670
18	1	5	3	6	24.2284	40.5168
19	4	5	6	7	24.4117	40.6084
20	3	2	6	4	24.5709	40.6874

TABLE VIII
Scene 4—Pairwise TTA Likelihood Table

	Sensor B							
Sensor A	1	2	3	4	5	6	7	0
1	0.8928	0.0475	0.0403	0.0194	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.6438	0.0000	0.0000	0.3562	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.5485	0.1174	0.0000	0.3340	0.0000	0.0000
4	0.0000	0.0000	0.0000	0.4122	0.1255	0.0752	0.3871	0.0000
0	0.1072	0.3087	0.4111	0.4509	0.5184	0.5908	0.6129	0.0000

to weight the hypotheses in some logical fashion. For example, hypothesis 1 out of 20 could get a weight of 20, hypothesis 2 a weight of 19,…, hypothesis 20 a weight of 1. As another example, one could use the GNPM likelihoods as weights. Finally, one could sum the total weights of the hypotheses in which tracks i and j are paired and normalize. Empirically, we have seen that employing a non-uniform weighting scheme often works reasonably well, and so the next logical question is: What is the correct weighting scheme? Indeed, in this paper, we argue that the MTTA likelihood values provide the "best" weights for the hypotheses. Moreover, while the ad hoc approaches in this second option may work, they lack probabilistic rigor. The MTTA likelihood has a solid mathematical backbone and, as these examples illustrate, furnishes insightful information for the system-user.

In summary, the above example illustrates the benefits of providing the system-user with several representations of the information that will be needed to assess the current situation and act appropriately. Although somewhat contrived, these vignettes clearly convey how both sensor bias and ambiguity could affect the results in a realistic tracking problem setting. We present both the GNPM and MTTA likelihood functions as measures for confidence in track-to-track association with sensor bias as well as illustrate how results will differ based on which one is chosen.

5. OPTIMIZING WITH RESPECT TO THE MTTA LIKELIHOOD

Equipped with a marginal association likelihood function that accounts for the presence of a random relative bias, it is natural to ask whether we can and should optimize track-to-track associations with respect to this likelihood function. In the last section, we attempted to provide a modestly compelling answer to the latter question regarding why one should at least consider using the association likelihood function for track-to-track association. In this section and the next, we answer the former question by describing how one can perform track-to-track association by optimizing with respect to the association likelihood function. In order to formalize this optimization problem, our objective in this section is to cast the track-to-track association problem, using the association likelihood function (8), as a mathematical program, specifically, as a 0-1 nonlinear optimization problem.

Starting from Equation (8), we collect the $(2\pi)^{D/2}$ terms and use the fact that $|I| + |I^0| = n_A$ to write

$$L(\mathbf{j}) = (\beta_{NTA}\beta_{NTB})^{n_A}(e^{(1/2)g})^{|I|}\frac{\sqrt{|\mathbf{V}|}}{\sqrt{\prod_{i\in I^+}|\mathbf{S}_{ij_i}|}}e^{-(1/2)\zeta}$$

where g is the gate value defined in Equation (3). Taking the logarithm and multiplying through by -2 yields

$$-2\log L(\mathbf{j}) = -2\log(\beta_{NTA}\beta_{NTB})^{n_A} - |I|g - \log(|\mathbf{V}|) + \sum_{i\in I^+}\log(|\mathbf{S}_{ij_i}|) + \zeta.$$

Replacing $\mathbf{S}_{0j_0}$ with $\mathbf{R}$ and ζ with $(\sum_{i\in I}\mathbf{x}_{ij_i}^T\mathbf{S}_{ij_i}^{-1}\mathbf{x}_{ij_i}) - \mathbf{u}^T\mathbf{V}\mathbf{u}$, where $\mathbf{V} = (\mathbf{R}^{-1} + \sum_{i\in I}\mathbf{S}_{ij_i}^{-1})^{-1}$, we obtain

$$-2\log L(\mathbf{j}) = \kappa + |I^0|g - \log(|\mathbf{V}|) + \sum_{i\in I}(\mathbf{x}_i^T\mathbf{S}_{ij_i}^{-1}\mathbf{x}_{ij_i} + \log(|\mathbf{S}_{ij_i}|)) - \mathbf{u}^T\mathbf{V}\mathbf{u} \tag{10}$$

where $\kappa = -2\log(\beta_{NTA}\beta_{NTB})^{n_A} - n_A g + \log(|\mathbf{R}|)$.

As in the GNPM problem, we introduce binary decision variables y_{ij} such that $y_{ij} = 1$ if sensor A track i is assigned to sensor B track j (or possibly the dummy track $j = 0$), and $y_{ij} = 0$ otherwise, as well as the set $\mathcal{Y}$ of all feasible associations. Finally, since κ is independent of the track-to-track association made, we treat it as a constant and remove it from the likelihood function when we optimize.

The problem of finding an optimal track-to-track association over all possible relative bias values can now be formulated as the following constrained 0-1 optimization problem, which we will henceforth refer to simply as the *Marginal Track-to-Track Association problem* or MTTA problem for short:

$$\begin{aligned}\min_{\mathbf{y}}\quad & \mathbf{c}^T\mathbf{y} - \log(|\mathbf{V}(\mathbf{y})|) - (\mathbf{A}\mathbf{y})^T\mathbf{V}(\mathbf{y})(\mathbf{A}\mathbf{y}) \\ \text{s.t.}\quad & \mathbf{V}(\mathbf{y}) = \left(\mathbf{R}^{-1} + \sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\mathbf{S}_{ij}^{-1}y_{ij}\right)^{-1}, \qquad \mathbf{y}\in\mathcal{Y}\end{aligned} \tag{11}$$

where $\mathbf{A}$ is a $D \times n_A(n_B + 1)$ matrix of reals whose columns are such that

$$\mathbf{A}\mathbf{y} = \sum_{i=1}^{n_A}\sum_{j=0}^{n_B}\left\{\begin{array}{ll}\mathbf{S}_{ij}^{-1}(\mathbf{x}_i^A - \mathbf{x}_j^B) & \text{if} \quad j > 0 \\ \mathbf{0} & \text{if} \quad j = 0\end{array}\right\}y_{ij}$$

the coefficients

$$c_{ij} = c_{ij}(\mathbf{0}) = \begin{cases} d_{ij}^2(\mathbf{0}) + \log|\mathbf{S}_{ij}| & \text{if} \quad j \in N_B \\ g & \text{if} \quad j = 0\end{cases}$$

for $i \in N_A$, are given, and g is the gate value defined in Equation (3). The mathematical program (11) can be classified as a nonconvex nonlinear matching problem, which is a class of optimization problems that is generally difficult to solve to provable optimality. In contrast with the GNPM problem (4), which includes a continuous decision variable (vector) to model the relative bias that we hope to determine and, consequently, is formulated as a MINLP, the MTTA problem does not include any continuous decision variables.

6. SOLUTION METHODS

In this section, we discuss solution techniques for solving the MTTA problem. In general, the class of 0-1 nonlinear optimization problems is challenging computationally and there are very few commercially-available optimization software packages that can solve them to provable optimality. Instead, highly-tailored algorithms that exploit idiosyncrasies of the particular problem are required. The MTTA problem is no exception. The effort needed to identify a provably optimal solution (i.e., a track-to-track association hypothesis) grows exponentially in the number of tracks to be associated.

On the other hand, finding near optimal solutions (and, perhaps, an optimal solution) without any guarantee on the solution quality, is frequently possible. It can be shown that under certain conditions [23], which often arise given realistic data, the GNPM problem discussed in Section 2 provides a close approximation to the MTTA problem. As a consequence, one approach to generating near optimal solutions as quickly as possible is first to solve the GNPM problem by applying the multistart local search heuristics described in [5] and [25]. Essentially, these heuristics apply a fast iterative local search method from various starting points (hence the term "multistart") and find good bias-association hypotheses, $(\mathbf{b},\mathbf{j})$ pairs, to the GNPM problem. To recover the "pure" association likelihood of association hypothesis $\mathbf{j}$, one can solve Equation (8). Because these are heuristics, no optimality guarantee is provided. Nevertheless, these methods are extremely fast and have proven to be important when faced with real-time processing constraints.

Given that heuristics are already available for finding good association hypotheses, we turn to the question of solving the MTTA problem *exactly*, i.e., identifying a provably optimal solution or the K provably best solutions. Typical approaches for solving 0-1 nonlinear optimization problems exactly include branch-and-bound methods, cutting-plane algorithms, and hybrids of these two known as branch-and-cut methods. Another interesting approach discussed in [18, 26], which is worthy of further investigation, is based on a technique known as "lifting" for 0-1 optimization problems and has gained increasing attention in the optimization community.

Below, we describe two different approaches for solving the MTTA problem to provable optimality. In the first approach, we solve the MTTA problem using a branch-and-bound method in which at each node in the search tree we solve (not necessarily to provable optimality) a quadratic assignment problem relaxation of the problem. As this approach does not scale well as the number of tracks increases, we provide a second approach in which we reformulate the MTTA problem as a mixed-integer linear optimization problem, which has many more variables and many more constraints than the original MTTA formulation, but performs remarkably well for a much larger set of instances of practical interest.

In what follows, we will need the following notation:

- The partial orderings $\mathbf{M} \succ \mathbf{0}$ and $\mathbf{M} \succeq \mathbf{0}$ mean that a square matrix $\mathbf{M}$ is positive definite and positive semidefinite, respectively. Given another square matrix $\mathbf{N}$, $\mathbf{M} \succ \mathbf{N}$ and $\mathbf{M} \succeq \mathbf{N}$ mean that $\mathbf{M} - \mathbf{N} \succ \mathbf{0}$ and $\mathbf{M} - \mathbf{N} \succeq \mathbf{0}$, respectively.
- We write $\mathbf{G} \cdot \mathbf{H} = \sum_i \sum_j g_{ij} h_{ij}$ to denote the Frobenius inner product of matrices $\mathbf{G}$ and $\mathbf{H}$.

6.1. Approach 1: A Branch-and-Bound Framework with Quadratic Assignment Problem Relaxations

In this section, we describe a branch-and-bound method that uses bounds obtained from solving a variant of the quadratic assignment problem at each node in the search tree. Branch-and-bound methods have been developed for the GNPM problem [5, 25] and closely resemble the one described below. The key difference between these methods and the one below is the construction of a lower bound at each node.

A systematic way of finding an optimal solution or the K best solutions to the MTTA problem is via a divide-and-conquer procedure known as branch-and-bound. In this approach, one creates a search tree consisting of nodes at varying depths, where each node represents a partial or complete association. Branch-and-bound methods pervade the field of discrete optimization and are discussed in virtually every introductory textbook on the subject (see, e.g., [22]). They are founded upon the idea of exploring nodes (or partial associations) in the search tree in an intelligent manner so that not all associations need be examined. The term *branch* refers to the manner in which partial associations are constructed, i.e., the partitioning of the solution space into smaller and smaller subproblems. The term *bound* refers to deterministic bounds that are computed during the search process, which can be used to prune partial associations that cannot possibly lead to (i.e., be a parent of) an optimal solution. All children of a pruned node are said to be implicitly enumerated.

We now describe a non-traditional branching strategy that has worked well in practice in which a parent node may have more than two children.[3] Without loss of generality, we assume that sensor A tracks are to be associated in increasing order with sensor B tracks, i.e., first track $1 \in N_A$ must be assigned, then track $2 \in N_A$, and so on. The root node has no associations and is said to be at depth 0. From the root node, $n_B + 1$ branches are created giving rise to $n_B + 1$ nodes at depth 1, which represent the association of sensor A track 1

[3] In most commercial and non-commercial mixed-integer programming solvers, a parent node gives rise to at most two child nodes so that a so-called binary tree is maintained. Empirically, we have found that the branching scheme described here works very well for the GNPM and MTTA problems.

with all n_B sensor B tracks plus the null partial association $\{(1,0)\}$. From each node at depth 1 with an association of the form $\{(1,j_1)\}$ with $j_1 \in N_B$, n_B branches are created. Each of these nodes has a partial association of the form $\{(1,j_1),(2,j_2)\}$, where $j_1 = 1,\ldots,n_B$ and $j_2 \in \{0,1,\ldots,n_B\}\backslash\{j_1\}$. From the node with partial association $\{(1,0)\}$, n_B+1 branches are created since sensor A track 2 can be associated with any track in N_B plus the dummy track. This branching continues until a depth n_A is reached or until all nodes have been implicitly enumerated.

Of course, a primary goal in branch-and-bound is to avoid having to expand a node by determining in advance if it has the potential of leading to an optimal solution or one of the K best solutions. If it can be deduced that no child node of a node that is being considered for expansion can be better than the best complete association(s) found thus far in the search process, known as the incumbent solution(s), then the node can be pruned. Pruning is essential when solving large problem instances because it reduces the time and memory needed to explore the search tree. In our solution approach, we advocate finding K good incumbent solutions quickly by first solving the GNPM problem and then launching branch-and-bound to prove the optimality of these solutions or to find better solutions.

In order to compute a valid lower bound at each node in the search tree to be used for pruning, we solve (not necessarily optimally) a variant of the quadratic assignment problem (QAP). The QAP is one of the most studied and notoriously difficult optimization problems. For more details, we refer the reader to [4] and [29]. When stated as a facility location problem, the classical QAP is to assign n facilities to n locations such that the total interaction cost of all possible flow-distance products between the locations to which the facilities are assigned and the allocation costs of facilities to locations are minimized [29]. If we let $N = \{1,\ldots,n\}$ be a set of nodes, $\mathcal{A} = \{(i,j) \in N \times N\}$ be a set of arcs, then the Koopmans-Beckmann form of the classical QAP is

$$\min \quad \sum_{(i,j)\in\mathcal{A}} \sum_{(k,l)\in\mathcal{A}} d_{ijkl} x_{ij} x_{kl} - \sum_{(i,j)\in\mathcal{A}} f_{ij} x_{ij} \tag{12a}$$

$$\text{s.t.} \quad \sum_{j\in N} x_{ij} = 1, \qquad \forall i \in N \tag{12b}$$

$$\sum_{i\in N} x_{ij} = 1, \qquad \forall j \in N \tag{12c}$$

$$x_{ij} \in \{0,1\}, \qquad \forall (i,j) \in \mathcal{A} \tag{12d}$$

where x_{ij} are assignment variables analogous to the y_{ij} variables in the MTTA problem, and d_{ijkl} and f_{ij} are data. In the generalized QAP, the number of facilities and locations need not be the same. In addition, the equality constraints (12c) are replaced with the more general inequality constraints $\sum_{i\in M} a_{ij} x_{ij} \le b_j$, $\forall j \in N$, where a_{ij} and b_j are data. Thus, constraints (12b) and (12c) are replaced by

$$\sum_{j\in N} x_{ij} = 1, \qquad \forall i \in M$$

$$\sum_{i\in M} a_{ij} x_{ij} \le b_j, \qquad \forall j \in N$$

where M is the set of locations and N is the set of facilities. The constraint set $\mathcal{Y}$ for the MTTA problem is therefore a special case of that of the generalized QAP in which a_{ij} and b_j are 1 for all $i \in M$ and $j \in N$.

6.1.1. Deriving Lower Bounds via QAP Relaxations

The basic idea behind obtaining a QAP relaxation of the MTTA problem at each node is as follows: At each depth $d \in \{0,1,\ldots,n_A-1\}$ in the search tree, we replace the matrix $\mathbf{V}(\mathbf{y})$, which depends on the decision vector $\mathbf{y}$ and consequently introduces unwieldy nonlinearities, with a suitable chosen positive definite matrix $\mathbf{V}_d$ such that

$$\mathbf{V}_d^{-1} := \mathbf{R}^{-1} + \sum_{i=1}^{d} \sum_{j=1}^{n_B} \mathbf{S}_{ij}^{-1} y_{ij}.$$

This leads to the following special case of the Generalized QAP at each node:

$$z^{\text{QAP}} = -\log(|\mathbf{V}_d|) + \min \ \mathbf{c}^T\mathbf{y} - \mathbf{y}^T\mathbf{Q}_d\mathbf{y} \tag{13a}$$

$$\text{s.t.} \quad \mathbf{y} \in \mathcal{Y} \tag{13b}$$

where $\mathbf{Q}_d = \mathbf{A}^T\mathbf{V}_d\mathbf{A}$. Notice that after replacing the matrix $\mathbf{V}(\mathbf{y})$ with a fixed matrix $\mathbf{V}_d$, the term $-\log(|\mathbf{V}_d|)$ becomes a constant and can be removed from the minimization. In what follows, we refer to the minimization problem in (13) as NodeQAP. In addition, note that NodeQAP is a special case of the generalized QAP stated as a maximization problem rather than as a minimization problem, which is more typical.

We now justify that this approach of replacing the matrix $\mathbf{V}(\mathbf{y})$ with the matrix $\mathbf{V}_d$ and solving NodeQAP provides a valid lower bound at each node in the search tree. The following proposition leads directly to the justification.

PROPOSITION 1 *For all depths $d \in \{0,1,\ldots,n_A-1\}$, define $\mathbf{V}_d^{-1} := \mathbf{R}^{-1} + \sum_{i=1}^{d}\sum_{j=1}^{n_B} \mathbf{S}_{ij}^{-1} y_{ij}$. Then,*

$$\mathbf{V}_{d+1}^{-1} \succeq \mathbf{V}_d^{-1} \qquad \textit{and} \qquad -\log|\mathbf{V}_{d+1}| \ge -\log|\mathbf{V}_d|.$$

PROOF Since the matrices $\mathbf{S}_{ij}$ and $\mathbf{S}_{ij}^{-1}$ are positive definite for all $i \in N_A$ and $j \in N_B$, it follows that $\mathbf{V}_{d+1}^{-1} = \mathbf{V}_d^{-1} + \mathbf{S}_{d+1,j}^{-1} y_{d+1,j} \succeq \mathbf{V}_d^{-1}$ for $d = 0,\ldots,n_A-1$ and for all $j \in \{0,\ldots,n_B\}$. Moreover,

$$\begin{aligned}
\mathbf{V}_{d+1}^{-1} \succeq \mathbf{V}_d^{-1} &\iff \mathbf{V}_{d+1} \preceq \mathbf{V}_d \\
&\iff |\mathbf{V}_{d+1}| \le |\mathbf{V}_d| \\
&\iff \log|\mathbf{V}_{d+1}| \le \log|\mathbf{V}_d| \\
&\iff -\log|\mathbf{V}_{d+1}| \ge -\log|\mathbf{V}_d|
\end{aligned}$$

where the first equivalence is a linear algebra exercise and the second equivalence holds because the determinant of the sum of two positive definite matrices is always greater than the determinants of either of those matrices.

In words, the proposition tells us that the matrix $\mathbf{V}_{d+1}^{-1}$ associated with a child node is at least as positive definite as that of its parent, and can only become more positive definite than that of its parent if sensor A track $d+1$ is assigned to a sensor B track $j \in N_B$. The implication of the proposition is: Since

$$\mathbf{y}^T\mathbf{A}^T\mathbf{V}_d\mathbf{A}\mathbf{y} \geq \mathbf{y}^T\mathbf{A}^T\mathbf{V}_{d+1}\mathbf{A}\mathbf{y}, \qquad \forall \mathbf{y} \in \mathcal{Y}\ ^4$$

and $-\log|\mathbf{V}_{d+1}| \geq -\log|\mathbf{V}_d|$, the optimal value z^{QAP} of (13) at a parent node at depth d will always be no greater than the optimal value at one of its descendants at depth $d+1, d+2, \ldots, n_A$. Therefore, z^{QAP} is a valid lower bound for the optimal objective function value of a node at depth d in the search tree.

6.1.2. Solving NodeQAP

There are several issues to consider when solving NodeQAP: How much time should be spent computing a lower bound at each node? Which formulation to use? Which algorithm to use?

We start with the first question. Our main goal of using a QAP relaxation at each node is to obtain as quickly as possible a good lower bound to facilitate pruning. Thus, it is not necessary to solve NodeQAP to optimality at every node since we simply need to determine whether or not this node has the potential to lead to one of the K best solutions. One is then faced with the following trade-off:

- Option 1: Spend a small amount of time at each node so that many nodes can be processed quickly, while possibly sacrificing our ability to prune nodes early in the tree;
- Option 2: Spend more time computing the best possible lower bound so that fewer nodes need to be expanded.

Since the classical QAP is a very difficult problem in its own right, empirically we have found it best to obtain a lower bound on NodeQAP at nodes with a small depth while forgoing optimality. Specifically, in our computational experiments, we set a time limit of a fraction of a second to obtain a lower bound at each node. We observed that even solving NodeQAP to provable optimality at a small depth (e.g., $d \leq n_A/4$) in the search tree rarely allows us to prune nodes. Consequently, it makes sense not to waste computational effort computing a bound that will not be immediately useful. On the other hand, as we dive deeper in the search tree, the lower bound furnished by (13) becomes stronger and more useful, while also becoming easier to solve to optimality since more assignments are fixed.

Regarding the question of which formulation to use, we have experimented with two approaches: reformulating NodeQAP as a mixed-integer linear program and relaxing NodeQAP as a semidefinite program. Semidefinite programming approaches are discussed in [4] and our initial semidefinite programming formulation is discussed in [24]; we will not discuss this approach here since, despite extensive experimentation, semidefinite programming software is not yet as mature as that for mixed-integer programming.

In our computations, we use the Kaufman-Broeckx linearization (1.12) on p.9 of [4], which is arguably the smallest linearization in terms of the number of variables and constraints in the model. The Kaufman-Broeckx formulation is a mixed-integer programming formulation of the QAP, which we solve using the commercially-available solver CPLEX 11.2 [13]. As discussed above, at shallow depths in the search tree, we have CPLEX compute the best lower bound for the Kaufman-Broeckx formulation of NodeQAP that it can provide in a fraction of a second (typically, by solving the linear programming relaxation of the Kaufman-Broeckx formulation of NodeQAP and performing a partial branch-and-bound). Later, at deeper depths in the search tree, CPLEX can solve the Kaufman-Broeckx formulation of NodeQAP to provable optimality in under a second.

Another important algorithmic enhancement is to recognize that it is not necessary to solve NodeQAP from scratch at each node. Indeed, information from the solution at a parent node should be exploited to help solve the QAP at a child node. In particular, there are two items that change from a parent node at depth d to a child node at depth $d+1$: the matrix $\mathbf{A}^T\mathbf{V}_d\mathbf{A}$ of the quadratic term in the objective function becomes $\mathbf{A}^T\mathbf{V}_{d+1}\mathbf{A}$ and an additional assignment constraint $y_{d,j_d} = 1$ appears in the child node QAP. At every node, we store the best lower bound and the best feasible solution computed in the allotted time limit. We then use this information to warm-start the solution procedure at each child node. Most mixed-integer programming solvers have options for warm-starting an algorithm using one or more known feasible solutions (see, e.g., [13]).[5]

6.1.3. A Branch-and-Bound Algorithm

We conclude this subsection with a high-level description of our branch-and-bound algorithm and with a proof of its correctness. Pseudocode is provided in Algorithm 1. Let $z(\mathbf{y}) = \mathbf{c}^T\mathbf{y} - \log|\mathbf{V}(\mathbf{y})| - (\mathbf{A}\mathbf{y})^T\mathbf{V}(\mathbf{y})(\mathbf{A}\mathbf{y})$. Let $z^{\text{QAP}}(\texttt{node})$ be the value of z^{QAP} in (13) at a particular node, call it `node`, in the search tree. Recall that at a

[4] Actually, this inequality holds for all $\mathbf{y} \in \mathbb{R}^{n_A(n_B+1)}$.

[5] In fact, it is precisely our ability to re-optimize NodeQAP at a child node given the solution at its parent that makes the mixed-integer programming formulation more attractive than the semidefinite programming formulation. Current semidefinite programming software does not have this algorithmic enhancement available.

node at depth $d \in \{0,1,\ldots,n_A\}$ in the search tree, the assignments of first d sensor A tracks are fixed. As is typical in a branch-and-bound implementation, we maintain a priority queue `pq` of nodes, whose `push(node)` method places `node` on the queue and whose `pop()` method (sometimes called a `findMin()` method) returns and removes the node with a minimum $z^{\text{QAP}}(\texttt{node})$ value from the queue. Note that this is a *best-first search* implementation and that the $z^{\text{QAP}}(\texttt{node})$ value of the node returned by the `pop()` method always provides a lower bound on the objective function value of the Kth provably best solution. Finally, the algorithm makes use of another method `createBranches(node,pq)`, which takes a given non-leaf node, creates its children as described in the beginning of this subsection, and adds these descendant nodes to the priority queue.

ALGORITHM 1 `BranchAndBoundforMTTA`(K)

1: Find K high-quality incumbent solutions $\mathbf{y}^1,\ldots,\mathbf{y}^K$ with $z(\mathbf{y}^1) \le \cdots \le z(\mathbf{y}^K)$
2: Initialize a priority queue of nodes: `pq={}`
3: Set $z^{UB} = z(\mathbf{y}^K)$; `pq.push(rootnode)`
4: **while** `pq` is not empty **do**
5: `node=pq.pop()`
6: **if** `node` is a leaf node **then**
7: Let $\hat{\mathbf{y}}$ be the solution associated with `node`
8: **if** $z(\hat{\mathbf{y}}) < z^{UB}$ and $\hat{\mathbf{y}} \notin \{\mathbf{y}^1,\ldots,\mathbf{y}^K\}$ **then**
9: Update list of K best solutions to include $\hat{\mathbf{y}}$
10: Update z^{UB}: $z^{UB} = z(\mathbf{y}^K)$
11: **else**
12: Prune `node`
13: **end if**
14: **else if** $z^{\text{QAP}}(\texttt{node}) > z^{UB}$ **then**
15: Prune `node`
16: **else**
17: `createBranches(node,pq)`
18: **end if**
19: **end while**
20: **return** K best track-to-track association solutions

THEOREM 1 (Algorithm 1 is exact) *The proposed branch-and-bound procedure finds the K provably best (globally optimal) solutions to the MTTA problem* (11).

PROOF The proof is by contradiction. Suppose Algorithm 1 returns the solutions $\mathbf{y}^1,\ldots,\mathbf{y}^K$ with $z(\mathbf{y}^1) \le \cdots \le z(\mathbf{y}^K)$. Suppose, to arrive at a contradiction, that there exists some solution $\hat{\mathbf{y}} \neq \mathbf{y}^k$, for $k = 1,\ldots,K$, such that $z(\hat{\mathbf{y}}) < z(\mathbf{y}^K)$. There are only two possible ways for (the leaf node corresponding to) the solution $\hat{\mathbf{y}}$ to have been pruned: implicitly or explicitly. If $\hat{\mathbf{y}}$ was pruned implicitly, then one of its ancestors in the search tree, call this node `ancestor`, had to be pruned in Step 15 because $z^{\text{QAP}}(\texttt{ancestor}) > z^{UB}$. Note that $z^{UB} \ge z(\mathbf{y}^K)$ since z^{UB} is an upper bound on the objective function value of the Kth best solution. But this immediately implies a contradiction since

$$z^{\text{QAP}}(\texttt{ancestor}) \le z(\hat{\mathbf{y}}) < z(\mathbf{y}^K) \le z^{UB} < z^{\text{QAP}}(\texttt{ancestor})$$

where the first inequality follows from Proposition 1 and the discussion that followed. On the other hand, if $\hat{\mathbf{y}}$ was pruned explicitly, then $\hat{\mathbf{y}}$ was pruned in Step 12. But this too implies a contradiction since, by assumption, $z(\hat{\mathbf{y}}) < z(\mathbf{y}^K)$. Hence, Algorithm 1 never incorrectly prunes a node, and therefore finds the K certifiably best solutions to the MTTA problem.

6.2. Approach 2: A Branch-and-Cut Algorithm for a MIP Reformulation

For our second approach, we reformulate the MTTA problem as a mixed-integer linear program (MIP) and then solve it using a branch-and-cut algorithm with a commercially-available MIP solver. A branch-and-cut algorithm is a more advanced branch-and-bound algorithm in which valid inequalities, known as "cutting planes" or simply as "cuts," are appended to the original formulation at each node in order to strengthen the formulation, improve bounds, and expedite node pruning. Virtually all commercial MIP solvers use a branch-and-cut algorithm rather than a pure branch-and-bound algorithm (see, for example, [13]).

Arriving at our proposed MIP reformulation takes two steps. In the first step, we introduce additional continuous auxiliary decision variables to reformulate the MTTA problem as a MINLP which possesses a special structure. Namely, the objective function consists of a submodular set function and a linear function, while the feasible region is a mixed-integer linear set, i.e., a set comprised of continuous and integer decision variables and linear inequalities. In other words, the only nonlinearity appears in the form of a submodular set function. In the second step, using well-known techniques for submodular function minimization, we replace the submodular function with a continuous auxiliary decision variable bounded below by a finite number of linear inequalities. This substitution gives rise to the desired MIP formulation, albeit one with a factorial number (in the number of tracks) of inequalities. Finally, we show how to generate only those cuts that are necessary in the formulation on an "as-needed" basis. Empirically, we have found this approach to be far superior to the branch-and-bound approach described above.

6.2.1. Towards a MINLP Reformulation

As outlined above, our first step towards achieving the desired MIP model is to reformulate the MTTA problem as a MINLP, which is a mixed-integer linear model except for a submodular function appearing in the objective function. To this end, note that the term $\mathbf{y}^T\mathbf{A}^T\mathbf{V}(\mathbf{y})\mathbf{A}\mathbf{y}$ in the objective function of (11) appears to be quadratic in $\mathbf{y}$, but is not because the matrix $\mathbf{V}(\mathbf{y})$ is the *inverse* of a sum of positive definite matrices

which depend on the decision variable $\mathbf{y}$. Our first goal is to circumvent the need to take an inverse by linearizing this term into a form suitable for a MIP solver. Consequently, we introduce an auxiliary decision vector $\mathbf{z} = \mathbf{V}(\mathbf{y})\mathbf{A}\mathbf{y}$ such that $\mathbf{V}^{-1}(\mathbf{y})\mathbf{z} = \mathbf{A}\mathbf{y}$. Recall that the inverse exists since all matrices are assumed to be positive definite. Then,

$$\begin{aligned}\mathbf{A}\mathbf{y} &= \mathbf{R}^{-1}\mathbf{z} + \sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\mathbf{S}_{ij}^{-1}\mathbf{z}y_{ij} \\ &= \mathbf{R}^{-1}\mathbf{z} + \sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\mathbf{S}_{ij}^{-1}\mathbf{w}_{ij}\end{aligned}$$

where $\mathbf{w}_{ij}$ is a D-dimensional vector of auxiliary decision variables such that

$$\begin{aligned}\mathbf{w}_{ij} &= \mathbf{z}y_{ij} \\ &= \begin{cases}\mathbf{z} & \text{if} \quad y_{ij} = 1, \\ \mathbf{0} & \text{if} \quad y_{ij} = 0,\end{cases} \qquad \forall i \in N_A, \quad j \in N_B \cup \{0\}\end{aligned}$$

and $\mathbf{w}_{i0} = \mathbf{0}$, $\forall i \in N_A$. Furthermore, we can linearize the nonlinear equations $\mathbf{w}_{ij} = \mathbf{z}y_{ij}$ (recall that both $\mathbf{z}$ and y_{ij} are decision variables) by replacing them with the following linear constraints:

$$-My_{ij}\mathbf{1} \le \mathbf{w}_{ij} \le My_{ij}\mathbf{1} \tag{14a}$$

$$-M(1-y_{ij})\mathbf{1} \le \mathbf{z} - \mathbf{w}_{ij} \le M(1-y_{ij})\mathbf{1} \tag{14b}$$

for all $i \in N_A$ and $j \in N_B$, where M is an appropriately chosen parameter and $\mathbf{1}$ is a D-dimensional vector of ones. This type of linearization of bi-linear terms is a standard "trick" in discrete optimization. Moreover, the nonlinear term $\mathbf{y}^T\mathbf{A}^T\mathbf{V}(\mathbf{y})\mathbf{A}\mathbf{y}$ in the objective function becomes $\mathbf{A}\cdot\mathbf{W}$, where $\mathbf{W} \in \mathbb{R}^{D\times n_A\times(n_B+1)}$ is the matrix satisfying $\mathbf{A}\cdot\mathbf{W} = \mathbf{y}^T\mathbf{A}^T\mathbf{z}$. This done, we have arrived at the following MINLP reformulation of the MTTA problem:

$$\min \quad f(\mathbf{y}) + \mathbf{c}^T\mathbf{y} - \mathbf{A}\cdot\mathbf{W} \tag{15a}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{y} = \mathbf{R}^{-1}\mathbf{z} + \sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\mathbf{S}_{ij}^{-1}\mathbf{w}_{ij} \tag{15b}$$

$$\text{(14a), (14b)}, \quad \mathbf{w}_{i0} = \mathbf{0}, \quad \forall i \in N_A, \quad \forall j \in N_B \tag{15c}$$

$$\mathbf{w}_{ij} \in \mathbb{R}^D, \quad \forall i \in N_A, \quad \forall j \in N_B \cup \{0\} \tag{15d}$$

$$\mathbf{y} \in \mathcal{Y}, \quad \mathbf{z} \in \mathbb{R}^D \tag{15e}$$

where $f(\mathbf{y}) := -\log(|\mathbf{V}(\mathbf{y})|) = \log(|\mathbf{V}^{-1}(\mathbf{y})|) = \log(|\mathbf{R}^{-1} + \sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\mathbf{S}_{ij}^{-1}y_{ij}|)$. To reiterate, with the exception of the nonlinear function $f(\mathbf{y})$ in the objective function, the above formulation is a MIP.

6.2.2. Towards a MIP Reformulation

The second step in obtaining the desired MIP reformulation is to show that the function $f(\mathbf{y})$ is a submodular set function and then describe how to exploit this submodularity in a branch-and-cut algorithm. Submodular functions have undergone extensive study in the optimization and computer science communities. We refer the reader to [1] and the references therein.

In what follows, let $n = n_A(n_B + 1)$ and define the set $N := \{1,\dots,n\}$. We also follow the common practice in the submodular function literature of abusing notation so that we may refer to a set function $h(S)$, for some $S \subseteq N$, as $h(\mathbf{x})$, where $\mathbf{x} \in \{0,1\}^n$ is the indicator vector of subsets of N. In addition, let $\mathbf{x}(S)$ denote the indicator vector of a set $S \subseteq N$, i.e., $x_i(S) = 1$ if $i \in S$, and 0 otherwise, for each component $i = 1,\dots,n$. Let $S_{\mathbf{x}}$ denote the support of a vector $\mathbf{x}$, i.e., $i \in S_{\mathbf{x}}$ if $x_i = 1$ and $i \notin S_{\mathbf{x}}$ if $x_i = 0$.

DEFINITION 1 A set function $h : 2^N \mapsto \mathbb{R}$ is *submodular* on N if

$$h(S) + h(T) \ge h(S\cup T) + h(S\cap T), \qquad \forall S,T \subseteq N.$$

A submodular function can also be characterized by its difference function $\rho_k(S) := h(S\cup\{k\}) - h(S)$ for $S \subseteq N$ and $k \in N\backslash S$. It is easy to verify that a set function h is submodular if and only if its difference function is nonincreasing; that is, $\rho_k(S) \ge \rho_k(T)$, $\forall S \subseteq T \subseteq N$ and all $k \in N\backslash T$. The following proposition shows that the function $f(\mathbf{y})$ is submodular on N by validating that the difference function of h is nonincreasing.

PROPOSITION 2 *Let* $N = \{1,\dots,n\}$ *and suppose that* $\mathbf{A}_i \succ \mathbf{0}$, $\forall i \in N \cup \{0\}$. *Then, the function* $g(S) := \log(|\mathbf{A}_0 + \sum_{j\in S}\mathbf{A}_j|)$ *is submodular on* N.

PROOF Let $S \subseteq T \subseteq N$ and $k \in N\backslash T$. Let $\mathbf{B} = \mathbf{A}_0 + \sum_{j\in S}\mathbf{A}_j$, $\mathbf{C} = \mathbf{A}_0 + \sum_{j\in T}\mathbf{A}_j$, and note that $\mathbf{C} \succ \mathbf{B}$ and $\mathbf{B}^{-1} \succ \mathbf{C}^{-1}$. Then,

$$\begin{aligned}g(S\cup\{k\}) - g(S) &= \log(|\mathbf{B} + \mathbf{A}_k|) - \log(|\mathbf{B}|) \\ &= \log\left(\frac{|\mathbf{B} + \mathbf{A}_k|}{|\mathbf{B}|}\right) \\ &= \log(|\mathbf{I} + \mathbf{A}_k\mathbf{B}^{-1}|) \\ &\ge \log(|\mathbf{I} + \mathbf{A}_k\mathbf{C}^{-1}|) \\ &= g(T\cup\{k\}) - g(T)\end{aligned}$$

where the inequality follows from the fact that $\mathbf{B}^{-1} \succ \mathbf{C}^{-1}$ and multiplying both sides by a positive definite matrix does not affect this ordering, nor does adding a positive definite matrix to both sides.

Since f satisfies the conditions of Proposition 2, f is submodular on N. We now present some well-known results from the theory of submodular function minimization, which will allow us to reformulate the MINLP as a MIP. The majority of the background material below is taken from Section 2 of Atamtürk and Narayanan [1]. Our basic goal is to replace the problem of minimizing a submodular set function $h(S)$ over all possible subsets S of N with an equivalent optimization problem of minimizing an auxiliary continuous variable

θ, representing $h(S)$, over the convex lower envelope of h, which is a polyhedron whose linear inequality description is already known to us due to the fact that h is submodular. With these linear inequalities in hand, we can transform our MINLP into a MIP, which can be solved by a MIP solver.

Suppose we are trying to minimize a submodular set function $h : 2^N \mapsto \mathbb{R}$ over the set N:

$$\min\{h(S) : S \subseteq N\}. \tag{16}$$

This is equivalent to

$$\min\{\theta : \theta \geq h(\mathbf{x}),\ \mathbf{x} \in \{0,1\}^n\}$$

or

$$\min\{\theta : (\theta, \mathbf{x}) \in \mathcal{Q}_h\} \tag{17}$$

where

$$\mathcal{Q}_h := \text{conv}\{(\theta, \mathbf{x}) \in \mathbb{R} \times \{0,1\}^n : \theta \geq h(\mathbf{x})\}$$

is the convex hull of the epigraph of h, which Atamtürk and Narayanan [1] refer to as the convex lower envelope of h; $\mathcal{Q}_h$ is also a polyhedron.[6] Without loss of generality, we may assume that $h(\emptyset) = 0$, since otherwise we can solve the equivalent minimization problem for the shifted function h' with $h'(S) := h(S) - h(\emptyset)$, $\forall S \subseteq N$.

It turns out that the linear inequalities describing $\mathcal{Q}_h$ are intimately related to the points in the *extended polymatroid* associated with h, which is defined as

$$EP_h := \{\pi \in \mathbb{R}^n : \pi(S) \leq h(S),\ \forall S \subseteq N\}$$

where $\pi(S) := \sum_{i \in S} \pi_i$. Atamtürk and Narayanan [1] formalize this relationship between the points in EP_h and the valid inequalities for $\mathcal{Q}_h$ in the following proposition.

PROPOSITION 3 (Atamtürk and Narayanan [1]) *The inequality $\pi^T \mathbf{x} \leq \theta$ is valid for $\mathcal{Q}_h$ (i.e., $\pi^T \mathbf{x} \leq \theta$, $\forall (\theta, \mathbf{x}) \in \mathcal{Q}_h$) if and only if $\pi \in EP_h$.*

PROOF Given $\pi \in EP_h$, we have $\pi^T \mathbf{x} = \pi(S_\mathbf{x}) \leq h(S_\mathbf{x}) \leq \theta$. Conversely, if $\pi \notin EP_h$, then $\pi(S) > h(S)$ for some $S \subseteq N$; but then for $\theta = h(S)$, $\pi(S) = \pi^T \mathbf{x}(S) > \theta$, contradicting the validity of $\pi^T \mathbf{x} \leq \theta$.

A fundamental goal in integer programming is to identify which are the important linear inequalities that are necessary in describing a polyhedron or a set of integer points and, therefore, provide the tightest formulation possible; these "important" inequalities are known as *facets* [22]. We call the bound constraints $\mathbf{0} \leq \mathbf{x} \leq \mathbf{1}$ *trivial* facets of $\mathcal{Q}_h$ since they can be shown to be facets of $\mathcal{Q}_h$ and are obtained trivially from relaxing the constraint $\mathbf{x} \in \{0,1\}^n$ to $\mathbf{x} \in [0,1]^n$. Although Proposition 3 allows us to determine whether or not a linear inequality is valid for $\mathcal{Q}_h$, it does not tell us if the inequality is a nontrivial facet of $\mathcal{Q}_h$. The next proposition allows us to identify nontrivial facets of $\mathcal{Q}_h$ and gives us an important relationship between the extreme points of the extended polymatroid and the convex hull of the epigraph. Since the proof given below was provided in an earlier version of [1], but does not appear in their published version, we state the proof for completeness.

PROPOSITION 4 *The inequality $\pi^T \mathbf{x} \leq \theta$ is a nontrivial facet for $\mathcal{Q}_h$ if and only if π is an extreme point of EP_h.*

PROOF $\Rightarrow$ From Proposition 3, if $\pi \notin EP_h$, inequality $\pi^T \mathbf{x} \leq \theta$ is not valid for $\mathcal{Q}_f$. If $\pi \in EP_h$ is not an extreme point of EP_h, then $\pi = \lambda \pi^1 + (1-\lambda)\pi^2$ for some $\lambda \in (0,1)$ and distinct points $\pi^1, \pi^2 \in EP_h$ and $\pi^T \mathbf{x} \leq \theta$ is implied by $(\pi^1)^T \mathbf{x} \leq \theta$ and $(\pi^2)^T \mathbf{x} \leq \theta$.

$\Leftarrow$ Conversely, if π is an extreme point of EP_h, it is the unique solution to a set of n linearly independent equations $\pi(S_i) = h(S_i)$ for $i = 1, \ldots, n$. Then, the corresponding linearly independent points $(\mathbf{x}(S_i), h(S_i))$, for $i = 1, \ldots, n$ of $\mathcal{Q}_h$ and the origin $(\mathbf{0}, 0)$ are on the face $\{(\mathbf{x}, \theta) \in \mathcal{Q}_h : \pi^T \mathbf{x} = \theta\}$. Finally, as $(\mathbf{0}, 1) \in \mathcal{Q}_h$, but is not on the face, the face is proper.

REMARK 1 Note that if $h(\emptyset) \neq 0$, the nontrivial facets for Q_h take the form $\pi^T \mathbf{x} \leq \theta - h(\emptyset)$, $\forall \pi \in EP_{h'}$, where $h' := h - h(\emptyset)$. Consequently, since $f(\emptyset) = \log(|\mathbf{R}|)$, the nontrivial facets of Q_f are of the form $\theta \geq \log(|\mathbf{R}|) + \pi^T \mathbf{y}$, $\forall \pi \in EP_f$.

The significance of Proposition 4 is that it tells us almost precisely which inequalities are needed to define the polyhedron $\mathcal{Q}_h$; the only piece of information that is missing is how to compute the coefficients π_k of these facets. Fortunately, this is provided in the following important result due to Edmonds [6].

THEOREM 2 (Edmonds [6]) *The point π is an extreme point of EP_h if and only if $\pi_i = h(S_{(i)}) - h(S_{(i-1)})$, where $S_{(i)} = \{(1), (2), \ldots, (i)\}$, for $i = 1, \ldots, n$, and $S_0 = \emptyset$, for some permutation $((1), (2), \ldots, (n))$ of N.*

In words, Proposition 4 and Theorem 2 state that there are $n!$ nontrivial facets needed to describe the polyhedron Q_h, one for each permutation of the elements of N, and the precise value of each coefficient depends on the corresponding permutation. When specialized to the submodular function f, Proposition 4, Remark 1, and Theorem 2 lead immediately to the following corollary.

COROLLARY 1 *The $n!$ nontrivial facets of the polyhedron Q_f, which we refer to as extended polymatroid (EP) inequalities, are of the form*

$$\theta \geq \log(|\mathbf{R}|) + \sum_{k=1}^{n} \pi_k^\psi y_{\psi(k)}, \qquad \forall \psi \in \Psi_n \tag{18}$$

where Ψ_n is the set of all permutations of $\{1, \ldots, n\}$ and π_k^ψ is the kth coefficient in permutation $\psi = \{(1), \ldots, (n)\}$ of Ψ_n. More precisely, $\pi_k^\psi = f(S_{(k)}) - f(S_{(k-1)}) = \log(|\mathbf{R}^{-1} + \sum_{a=1}^{k} \mathbf{S}_{(a)}^{-1}|) - \log(|\mathbf{R}^{-1} + \sum_{a=1}^{k-1} \mathbf{S}_{(a)}^{-1}|)$, where a denotes the track pair (i, j) associated with the ath element of the permutation ψ.

[6] This fact is important as it assures us that $\mathcal{Q}_h$ can be described by a *finite* number of linear inequalities.

REMARK 2 If $\theta < f(\mathbf{y})$ and $\mathbf{y} \in [0,1]^n$, then, by definition, the point $(\theta, \mathbf{y}) \notin Q_f$ and hence, by Corollary 1, at least one EP inequality (18) is violated.[7]

Using the equivalence that allowed us to express (16) as (17), and by applying Corollary 1 to provide us with an explicit polyhedral representation of Q_f, we can transform Formulation (15) and pose the MTTA problem as the following MIP:

$$\min \quad \theta + \mathbf{c}^T\mathbf{y} - \mathbf{A}\cdot\mathbf{W} \tag{19a}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{y} = \mathbf{R}^{-1}\mathbf{z} + \sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\mathbf{S}_{ij}^{-1}\mathbf{w}_{ij} \tag{19b}$$

$$(14a),\ (14b), \quad \mathbf{w}_{i0} = \mathbf{0}, \quad \forall i \in N_A, \quad \forall j \in N_B \tag{19c}$$

$$\mathbf{w}_{ij} \in \mathbb{R}^D, \quad \forall i \in N_A, \quad \forall j \in N_B \tag{19d}$$

$$\mathbf{z} \in \mathbb{R}^D, \quad \mathbf{y} \in \mathcal{Y}, \quad \theta \geq \log(|\mathbf{R}|) \tag{19e}$$

$$\theta \geq \log(|\mathbf{R}|) + \sum_{k=1}^{n}\pi_k^{\psi} y_{\psi(k)}, \quad \forall \psi \in \Psi_n. \tag{19f}$$

In contrast to the original formulation of the MTTA problem (11) and the formulation of NodeQAP (13), both of which operate in the space ($\mathbb{R}^n$) of the original decision variables $\mathbf{y}$, Formulation (19) requires us to introduce $O(Dn)$ auxiliary decision variables, thus forcing us to operate in a higher-dimensional space, and $O(3n)$ additional constraints prior to introducing constraints (19f). At first glance, such a transformation may seem "indirect" and fruitless. However, the advantage of this formulation is that it is a MIP and can, in theory, be solved by an off-the-shelf MIP solver. Indeed, the benefits of working in a higher-dimensional space will become clear in Section 7.

6.2.3. A Branch-and-Cut Algorithm

Unfortunately, the above MIP formulation requires a factorial number ($n!$) of EP inequalities (19f), which is unwieldy for instances of practical interest. However, it turns out that this drawback can easily be overcome. Since very few of these EP constraints will be tight at an optimal solution, rather than include them all in the initial formulation, we omit all of them from the outset, and then generate those that are necessary on an "as-needed" basis. Specifically, we initially formulate the MIP (19a)–(19e). Then, at every node in the search tree, we check if $\hat{\theta} \geq f(\hat{\mathbf{y}}) = \log(|\mathbf{R}^{-1} + \sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\mathbf{S}_{ij}^{-1}\hat{y}_{ij}|)$ where $\hat{\theta}$ and $\hat{\mathbf{y}}$ are partial solutions (obtained from the linear programming relaxation) at that node. Note that $\hat{\mathbf{y}}$ will have fractional components at a non-leaf node in the search tree. If $\hat{\theta} < f(\hat{\mathbf{y}})$, then, by definition, $(\hat{\theta}, \hat{\mathbf{y}}) \notin Q_f$ and so we "cut off" the solution at that node by appending the most violated EP inequality, which is guaranteed to exist by Remark 2. Edmonds [6] showed that finding the most violated EP inequality can be solved very efficiently using "the Greedy Algorithm" 2.

ALGORITHM 2 `GreedyAlgorithm`$(f, \hat{\mathbf{y}})$

1: Sort the $\hat{y}_i$ variables in nonincreasing order, $\hat{y}_{[1]} \geq \cdots \geq \hat{y}_{[n]}$, breaking ties arbitrarily
2: Define $S_{[i]} := \{[1],[2],\ldots,[i]\}$, for $i = 1,\ldots,n$, and $S_0 := \emptyset$
3: Define $\pi_{[i]} := f(S_{[i]}) - f(S_{[i-1]})$, for $i = 1,\ldots,n$
4: **return** The EP inequality $\pi^T\mathbf{y} \leq \theta - f(\emptyset)$

A high-level sketch of the algorithm used to find the single provably best (optimal) solution to the MTTA problem (11) is outlined in Algorithm 3. The pseudocode is more terse than that given in Algorithm 1 because we have implemented our algorithm in a commercial solver (CPLEX [13]) which manages the branch-and-bound tree for the user and provides all of the necessary functionality for solving linear and mixed-integer programs. Algorithm 3 takes a single input parameter: `MIPmodel`. Initially, `MIPmodel` represents the MIP formulation (19a)–(19e). The only advanced technique that we employ is in telling the solver how to identify if a violated EP constraint (19f) exists. This step is implemented through a "callback" function, which virtually all of the leading commercial and open-source MIP solvers provide.

ALGORITHM 3 `BranchAndCutforMTTA` *(*`MIPmodel`*)*

1: Write a callback function to do the following at each node in the search tree:
2: **while** $\hat{\theta} < f(\hat{\mathbf{y}})$ **do**
3: Call `GreedyAlgorithm`$(f, \hat{\mathbf{y}})$ and append the returned EP inequality $\pi^T\mathbf{y} \leq \theta - f(\emptyset)$ to `MIPmodel`
4: Re-solve the LP relaxation of `MIPmodel` to produce an updated partial solution $(\hat{\theta}, \hat{\mathbf{y}})$
5: **end while**
6: Invoke the MIP solver's `solve()` method with the callback function
7: **return** The single best track-to-track association solution $\hat{\mathbf{y}}^*$ to `MIPmodel`

When solving a MIP, all leading MIP solvers relax the binary constraints $\mathbf{y} \in \{0,1\}^n$ with linear constraints $\mathbf{y} \in [0,1]^n$ so that a linear program (LP) is solved at every node in the search tree. Before branching on a decision variable that is required to be integral, but is currently fractional at the optimal solution of the LP relaxation, the solver will attempt to generate a number of default cuts that apply to all MIPs as well as those cuts that the user requested the solver to attempt to generate through a callback function. Thus, if in Step 2 the solver finds that $\hat{\theta} < f(\hat{\mathbf{y}})$, the most violated EP cut (19f) is appended to `MIPmodel` and the LP relaxation of

[7] Up to this point, we have described f as a set function, i.e., as a function $f(S)$ whose argument is a subset S of N or, equivalently, as a function $f(\mathbf{y})$ whose argument is a binary vector $\mathbf{y} \in \{0,1\}^n$. Henceforth, when we write $f(\mathbf{y})$, we will treat f as a continuous function of $\mathbf{y} \in [0,1]^n$.

MIPmodel is re-solved in Step 4 for an updated partial solution $(\hat{\theta}, \hat{\mathbf{y}})$. In the event that the node being processed is a leaf node, which implies that $\hat{\mathbf{y}}$ is binary, the solver will still call the callback function (Step 2) to check for violated EP inequalities. This loop is repeated at every node until no more violated EP inequalities are found. When Algorithm 3 has terminated, MIPmodel will likely have a very small subset of EP constraints as well as a number of default MIP cuts that were added by the solver. This is an important observation because we will re-use this updated MIPmodel when finding the K best solutions.

THEOREM 3 (Algorithm 3 is exact) *The proposed branch-and-cut procedure finds the provably best (globally optimal) solution to the MTTA problem* (11) *when the input parameter* MIPmodel *represents the MIP formulation* (19a)–(19e).

PROOF Let $(\theta^*, \mathbf{w}^*, \mathbf{y}^*, \mathbf{z}^*)$ be an optimal solution to (19) with corresponding objective function value γ^*. Let $(\theta^{**}, \mathbf{w}^{**}, \mathbf{y}^{**}, \mathbf{z}^{**})$ be the solution returned by Algorithm 3 with corresponding objective function value γ^{**}. Note that $\gamma^{**} \leq \gamma^*$since, upon termination, MIPmodel does not necessarily (and most likely does not) contain all EP constraints (19f) and is, therefore, a relaxation of (19). Meanwhile, $(\theta^{**}, \mathbf{w}^{**}, \mathbf{y}^{**}, \mathbf{z}^{**})$ is feasible to (19) since in Step 3 no more violated EP constraints could be found, which by Corollary 1 occurs if and only if $(\theta^{**}, \mathbf{y}^{**}) \in Q_f$. Thus, $(\theta^{**}, \mathbf{w}^{**}, \mathbf{y}^{**}, \mathbf{z}^{**})$ must be an optimal solution to (19) and the assignment $\mathbf{y}^{**}$is an optimal solution to the MTTA problem (11).

Thus far, we have described how to find the single provably best solution to the MTTA problem. In order to find the K provably best solutions, we essentially make K calls to Algorithm 3. A high-level description of the algorithm for finding the K provably best solutions to the MTTA problem is presented in Algorithm 4. In Step 1, we formulate MIPmodel as the MIP formulation (19a)–(19e). It is important to note that MIPmodel will be modified throughout the algorithm as other constraints are appended to it. Step 2 is not necessary, but is almost always a good idea since MIP solvers generally perform better when good incumbent solutions are already available. The main **while** loop in Steps 4–10 requires the most computation time. Initially, we find the kth best solution for $k = 1$. In general, after finding the kth best optimal solution $\hat{\mathbf{y}}^k$, we add a so-called *enumeration cut*

$$\sum_{j=1:\hat{y}_j^k=1}^{n} (1 - y_j) + \sum_{j=1:\hat{y}_j^k=0}^{n} y_j \geq 1 \tag{20}$$

to MIPmodel to render the current optimal solution infeasible and thus allow us to find the solution corresponding to the next best association. Re-solving MIPmodel in Step 5 starting from the formerly optimal, but now infeasible, solution will yield the next best provably optimal solution.

ALGORITHM 4 MIPforMTTA(K)

1: Formulate the MIP (19a)–(19e). Call this formulation MIPmodel.
2: Obtain high-quality incumbent solutions by solving the GNPM problem for the K best bias-assignment solutions $\hat{\mathbf{y}}^1, \ldots, \hat{\mathbf{y}}^K$ using a good heuristic
3: Set numSolnsFound=0
4: **while** numSolnsFound $< K$ **do**
5: Call BranchAndCutforMTTA (MIPmodel) for an optimal assignment $\hat{\mathbf{y}}^*$ to MIPmodel
6: Update the list of the K best solutions, if possible
7: numSolnsFound++
8: **if** numSolnsFound $< K$ **then**
9: Append an enumeration cut to MIPmodel: $\sum_{j=1:\hat{y}_j^*=1}^{n}(1 - y_j) + \sum_{j=1:\hat{y}_j^*=0}^{n} y_j \geq 1$
10: **end if**
11: **end while**
12: **return** The K best track-to-track association solutions $\hat{\mathbf{y}}^1, \ldots, \hat{\mathbf{y}}^K$

THEOREM 4 (Algorithm 4 is exact) *Algorithm 4 finds the K provably best (globally optimal) solutions to the MTTA problem* (11).

PROOF By induction. Let k = numSolnsFound. When $k = 0$, Algorithm 4 calls Algorithm 3 without any enumeration cuts appended to MIPmodel. By Theorem 4, the solution returned by Algorithm 3 without any enumeration cuts is a provably optimal solution to (19). For the inductive step, assume that at the start of Step 4 when $0 < k < K$, the k provably best associations $\mathbf{y}^1, \ldots, \mathbf{y}^k$ have been found and that MIPmodel includes k enumeration cuts (20) that render the k best associations infeasible. Since all solutions to the MIP (19a)–(19e) are still feasible to MIPmodel, except for those whose $\mathbf{y}$-component corresponds to one of the k best associations, the next call to BranchAndCutforMTTA(MIPmodel) will solve a MIP over a smaller feasible region and, by using arguments virtually identical to those given in the proof of Theorem 4, will return a solution whose $\mathbf{y}$-component corresponds to the $(k + 1)$th globally optimal association.

7. PERFORMANCE STUDIES

The purpose of this section is to compare the performance of the two exact algorithms described in the previous section. It will be shown that the branch-and-cut algorithm (Approach 2) is vastly superior to the branch-and-bound method (Approach 1). We will also see empirical evidence that the K best MTTA solutions are often a subset of the $3K$ best GNPM solutions. It is for this reason, as well as for the arguments given in Section 4 with the illustrative example, that we do not specifically evaluate the suitability of the MTTA likelihood function for performing track-to-track association in comparison to the GNPM likelihood function. The

TABLE IX
Density Guidelines

Overlap %	MMD	PCA
100%	[0,2)	[0,0.75)
	[2,4)	[0.75,0.95)
	[4,∞)	[0.95,1]
70%	[0,4)	[0,0.75)
	[4,8)	[0.75,0.95)
	[8,∞)	[0.95,1]

TABLE X
3D Cube Side Lengths

	Density		
n_B	High	Medium	Low
5	5.26	10.51	15.77
6	6.10	12.20	18.29
7	6.89	13.79	20.68
8	7.65	15.29	22.94
9	8.36	16.72	25.08
10	9.06	18.11	27.17
11	9.72	19.44	29.16
12	10.37	20.73	31.10
13	10.99	21.98	32.97
14	11.61	23.22	34.83
15	12.20	24.39	36.59

superiority of one over the other is an open question, although we contend that the MTTA likelihood function is more suitable in the case of a one-time handover.

7.1. Instance Generator

For ease of explanation and reproducibility for other researchers, we present computational results for 3-dimensional "box" instances of the MTTA problem. Although numerous experiments involving more realistic tracking scenarios in 6 dimensions have been completed, we believe that the same insights about the algorithms can be gleaned from these contrived instances.

Track states are maintained in a 3-dimensional Cartesian reference frame. All covariance matrices are diagonal matrices: $\mathbf{R} = \mathbf{P}_i = \mathbf{Q}_j = \mathbf{I}$ so that $\mathbf{S}_{ij} = 2\mathbf{I}$, $\forall i \in N_A$, $\forall j \in N_B$. The symmetry (i.e., the homogeneity of the covariances matrices) of these instances leads to difficulties for the algorithms as they cannot help distinguish between choosing one track pairing over another. So while these instances may appear simple, they are actually more difficult than most instances encountered in practice in which disparate covariances are typical.

A particular "box" instance is created as follows: After choosing n_A and n_B (with $n_A \leq n_B$), n_B objects are created by randomly generating position components uniformly in a cube with a given side length and assigned to sensor B. Next, n_A of the n_B objects are identified and assigned to sensor A. The true bias is drawn randomly from a Gaussian$(\mathbf{0},\mathbf{R})$ distribution and is added to each track in N_A. Finally, each track in N_A and N_B receives a random measurement error, drawn randomly from a Gaussian$(\mathbf{0},\mathbf{I})$ distribution.

The side length of the cube from which the position estimates of the tracks are randomly drawn influences the track scene density, i.e., how closely spaced the objects are to one another. In general, the more closely spaced are the tracks, the more difficult it is to correctly associate tracks (with respect to truth) and the more computational effort is required to find provably optimal solutions. While many metrics could be used to gauge scene density, one that has been used in a number of studies is the *minimum Mahalanobis positional distance* (MMD) computed over all tracks in each track set. It is a unitless metric sometimes referred to as the minimum nearest neighbor distance. Based on several computational studies, e.g., [25], [23], [7], Table IX was constructed to provide rough guidelines regarding the difficulty to correctly associate tracks in terms of the probability of correct association (PCA), defined to be the fraction of the n_A sensor A tracks that the association algorithm correctly associates based on truth. For example, when $n_A = n_B$ (100% overlap between the two track sets) and the MMD is less than 2, Table IX indicates that PCA is, on average, less than 0.75. Similar guidelines are given for an overlap of 70%, meaning $n_A = 0.7n_B$.

Using the above density guidelines, we evaluated the performance of our algorithms at three different scene densities: low, medium, and high. A high density scene corresponds to a MMD in the interval [1,2), a medium density in [2.5,3.5], and a low density in [4,5]. For low, medium, and high densities, we expected and confirmed that the average PCA was at least 0.95, between 0.75 and 0.95, and less than 0.75, respectively. To generate instances with these densities, we generated n_B points from a cube with the side length given in Table X. These parameters were computed through simulation and yield, on average, a MMD in the desired interval. If the MMD did not fall within the desired interval, this instance was discarded and new instances were generated until the MMD fell within the desired interval.

7.2. Computational Results

All computations were carried out on a Linux machine with kernel 2.6.18 running on a 64-bit x86 processor and 32GB of RAM. For every choice of parameter settings tested, 100 Monte Carlo experiments were performed. Both algorithms were implemented in Java and the MIP solver of CPLEX 11.2 [13] was used for the branch-and-cut algorithm. A solution was declared optimal if the relative optimality gap $(z^* - z^{LB})/z^*$ was at most 0.0001, where z^* is the value of the current best solution, and z^{LB} is a lower bound on the value of the MTTA problem. At the start of each exact algorithm, we employed the "All-Pairs" heuristic proposed in [5] to obtain K near-optimal solutions to the GNPM problem. While there are many heuristics available, we chose the

TABLE XI
Branch-and-Bound CPU Times (seconds)

# Tracks		Density		
n_A	n_B	Low	Medium	High
5	5	1.6981	1.8756	2.3923
6	6	16.5326	19.6353	25.1325
7	7	123.4889	131.6625	213.7652
8	8	287.4124	294.4444	—
9	9	—	—	—
10	10	—	—	—
5	10	60.7861	69.8828	87.3136
6	11	142.4322	155.6128	—
7	12	259.7773	274.3616	—
8	13	—	—	—
9	14	—	—	—
10	15	—	—	—

TABLE XII
Branch-and-Cut CPU Times (seconds)

# Tracks		Density		
n_A	n_B	Low	Medium	High
5	5	0.3599	0.4417	1.4738
6	6	0.4976	0.6367	1.5398
7	7	0.6778	0.7988	4.0195
8	8	0.9395	1.1405	7.1438
9	9	1.1608	1.4496	10.3954
10	10	1.5551	1.8486	23.5589
11	11	1.9391	2.2815	29.5981
12	12	2.6192	2.7251	31.3729
13	13	3.2803	3.3583	56.5361
14	14	4.1458	4.2619	74.1426
15	15	4.4676	4.8745	78.0808
5	10	0.4654	0.5369	1.5663
6	11	0.6787	0.7290	2.9092
7	12	0.8933	0.9331	4.2180
8	13	1.1667	1.2309	6.4942
9	14	1.5173	1.5502	10.0925
10	15	1.9460	1.9844	12.3069

"All-Pairs" heuristic for its simplicity and deterministic behavior. As will be explained in Fig. 5, we have empirical evidence that the GNPM likelihood function closely approximates the MTTA likelihood function (at least in the region of optimal solutions), and so the K near-optimal GNPM solutions returned by the heuristic are near-optimal for the MTTA problem. This is good news since it means that good heuristics already exist and new heuristics do not need to be developed. Since the heuristic is typically able to find a subset of the K best solutions, most of the effort carried out by the exact algorithms is in proving that these solutions are, indeed, optimal.

Tables XI and XII show the average solution times in seconds needed for the branch-and-bound and the branch-and-cut algorithms, respectively, to identify the $K = 10$ provably best solutions for various track sizes and scene densities. A dash (—) in the table means that, on average, an algorithm did not terminate within a five minute time limit. Even after significant experimentation with different parameter settings, the branch-and-bound method is vastly inferior to the branch-and-cut algorithm. Indeed, the branch-and-bound method cannot solve instances with more than 8 tracks in each set in under five minutes. Meanwhile, for low and medium scene densities, the branch-and-cut algorithm performs remarkably well with average solution times under five seconds. The standard deviation in solution time was also under a second for each parameter setting.

Although the solution times for the high density track scenes may appear discouraging, empirically we have observed that any association algorithm optimizing with respect to the GNPM or MTTA likelihood function will have a low PCA at such a high density. In fact, these track scenes are so congested that one is likely to question the fidelity of the tracks that were formed, i.e., the observation-to-track assignments that were made to produce the tracks are likely to be flawed. In light of this caveat, the solution times presented for the high density track scenes should be considered a "stress test" for the algorithm under extreme conditions. And if these conditions are encountered in practice, then one might verify that the track states and covariances are valid.

We attribute this marked discrepancy in performance between the two algorithms to two facts. First, in the branch-and-bound method, the QAP relaxations at shallow depths in the search tree are poor and do not permit early pruning, which leads to many nodes being expanded early in the search. Second, once the bounds do become useful for pruning, one still has to solve a small QAP, which may require roughly a second of computation time. Coupling these two facts, we see that many small QAPs, each requiring a small, but non-negligible amount of time, quickly adds up. On the other hand, the branch-and-cut method fully exploits the power and efficiencies that are now standard in MIP solvers, which leads to impressive computation times.

Another interesting question is: How well does the branch-and-cut algorithm scale as the user requests more and more provably optimal solutions, i.e., as the parameter K increases? Fig. 4 illustrates that, for various track sizes with a medium scene density, the algorithm scales almost linearly in K.

Fig. 5 shows the approximate number of provably optimal GNPM solutions that must be generated in order to ensure, with a high probability, that the K provably best solutions are obtained. From a theoretical perspective, this figure shows the close relationship between the GNPM and MTTA likelihood functions. From a practical perspective, this figure is important because it indicates that a user who does not wish to implement the branch-and-cut method, or any other exact algorithm for that matter, but who already has access to a good heuristic for the GNPM problem, can use the existing heuristic to generate optimal or near-optimal solutions with high

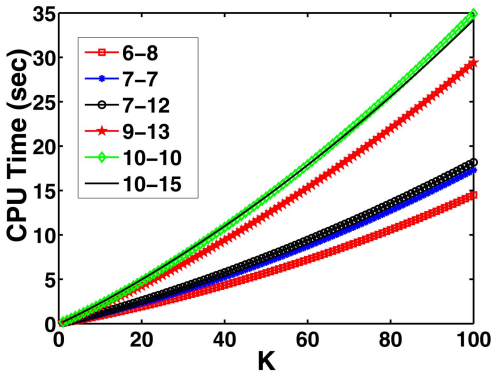


Fig. 4. Solution time (seconds) as a function of K.

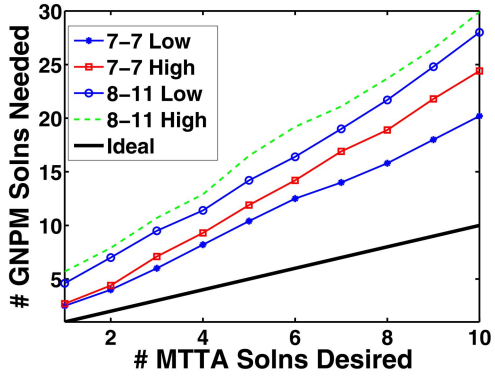


Fig. 5. Number of GNPM solutions needed.

confidence. Indeed, after choosing the number K of optimal MTTA solutions desired on the x-axis, the user can get a rough sense of the number of GNPM solutions that should be generated in order to be confident that the K best MTTA solutions have been found. Specifically, for a given choice of K, Fig. 5 shows the mean number (plus three standard deviations) of provably optimal GNPM solutions needed to obtain the K provably best MTTA solutions. Ideally, the K best MTTA hypotheses would coincide with the K best GNPM hypotheses in which case the fast multistart local search heuristics that are already available for the GNPM problem could be used for the MTTA problem without any loss of optimality. In this ideal setting, we would obtain the thick black line emanating from the origin. Unfortunately, the best hypotheses for the different likelihood function generally do not move in lock step.

Although we only present results up to the 10 best solutions, we have found that this pattern continues. Thus, we suggest that if a user wishes to obtain the K best MTTA solutions with high confidence without implementing one of our exact algorithms, he should find the $3K$ best GNPM solutions to achieve this goal.

At various times, we have used the superlative "remarkable" to describe the branch-and-cut algorithm. We close this section to explain why we believe the results for the branch-and-cut algorithm are so impressive. First, operational requirements for most association algorithms typically allow at most a few seconds of computation time. Thus, having the ability to return provably optimal solutions in such a small time window is a desirable feature. Second, since many heuristics for the GNPM problem require one- or two-tenths of a second to return a number of near optimal solutions, it appears that the increase in computation time to identify provably optimal solutions is roughly an order of magnitude. Given that the increase in solution time to find a single provably optimal solution over a heuristic solution is often several orders of magnitude for many difficult 0-1 optimization problems, having an order of magnitude increase is good news.

8. CONCLUSIONS

The primary goal of this paper was to introduce a marginal track-to-track association likelihood function for track ambiguity management, which takes into account all of the major issues considered by other popular association likelihood functions, but is more suitable for system-level track ambiguity management, especially for a one-time handover. We described how pairwise track-to-track likelihoods could be constructed to quantify the confidence in pairing two tracks together. Our final contribution was the introduction of two exact algorithms that can solve a track-to-track association problem using the likelihood function that we introduced. The second approach, which exploits well-known results from submodular function minimization, performs quite well.

As there is on-going effort to develop efficient and robust algorithms for performing track-to-track association between more than two sensors, we believe that understanding the possible types of information that can be extracted when only two sensors are participating and the shortcomings of the related algorithms to obtain this data are crucial when considering more complex multisensor problems. In addition, since data association is a low level step in the data fusion process, improving metrics and algorithms for track correlation can have a significant impact on system performance.

ACKNOWLEDGMENTS

We wish to thank Arjang Noushin, John-David Sergi, and Andy Stachyra of Raytheon for valuable feedback. The first author would like to thank Maxim Raykin of Raytheon for helping recognize the closed-form solution in Equation (8). We thank three anonymous referees for their insightful comments which helped improve the paper as a whole.

REFERENCES

[1] A. Atamtürk and V. Narayanan
Polymatroids and mean-risk minimization in discrete optimization.
Operations Research Letters, **36** (2008), 618–622.

[2] Y. Bar-Shalom and H. Chen
Track-to-track association using attributes.
Journal of Advances in Information Fusion, **2**, 1 (2007), 49–59.

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

[4] E. Çela
The Quadratic Assignment Problem: Theory and Algorithms.
Kluwer Academic Publishers, 1998.

[5] S. Danford, B. D. Kragel, and A. B. Poore
Joint bias estimation and data association: Algorithms.
In O. E. Drummond and R. D. Teichgraeber (Eds.), *Proceedings of SPIE*, vol. 6699, Signal and Data Processing of Small Targets, 2007.

[6] J. Edmonds
Submodular functions, matroids and certain polyhedra.
In R. Guy (Ed.), *Combinatorial Structures and Their Applications*, vol. 11, Gordon and Breach, New York, NY, 1971, 69–87.

[7] T. Fercho and D. J. Papageorgiou
Feature-aided global nearest pattern matching with non-gaussian feature measurement errors.
In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT: IEEE, Mar. 2009.

[8] J. Ferry
Exact bias removal for the track-to-track association problem.
In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, WA, July 2009.

[9] J. Ferry
Exact association probability for data with bias and features.
Journal of Advances in Information Fusion, **5**, 1 (2010), 41–67.

[10] D. L. Hall and J. Llinas
An introduction to multisensor data fusion.
Proceedings of the IEEE, **85**, 1 (1997), 1–18.

[11] R. E. Helmick and T. R. Rice
Removal of alignment errors in an integrated system of two 3-d sensors.
IEEE Transactions on Aerospace and Electronic Systems, **29**, 4 (1993), 1333–1343.

[12] S. M. Herman and A. B. Poore
Nonlinear least-squares estimation for sensor and navigation biases.
In O. E. Drummond (Ed.), *Proceedings of SPIE*, vol. 6236, Signal and Data Processing of Small Targets, Aug. 2006.

[13] ILOG CPLEX
Reference Manual, Version 11.2, 2008.

[14] R. J. Kenefic
Local and remote track file registration using minimum description length.
IEEE Transactions on Aerospace and Electronic Systems, **29**, 3 (1993), 651–655.

[15] B. D. Kragel, S. Danford, S. M. Herman, and A. B. Poore
Bias estimation using targets of opportunity.
In O. E. Drummond (Ed.), *Proceedings of SPIE*, vol. 6699, Signal and Data Processing of Small Targets, Aug. 2007.

[16] B. D. Kragel, S. Danford, and A. B. Poore
Concurrent map data association and absolute bias estimation with an arbitrary number of sensors.
In O. E. Drummond (Ed.), *Proceedings of SPIE*, vol. 6969, Signal and Data Processing of Small Targets, Aug. 2008.

[17] M. Levedahl
An explicit pattern matching assignment algorithm.
In O. E. Drummond (Ed.), *Proceedings of SPIE*, vol. 4728, Signal and Data Processing of Small Targets, Aug. 2002, 461–469.

[18] L. Lovász and A. Schrijver
Cones of matrices and set-functions and 0-1 optimization.
SIAM Journal on Optimization, **1** (1991), 166–190.

[19] J. R. Moore and W. D. Blair
Practical aspects of multisensor tracking.
In Y. Bar-Shalom and W. D. Blair (Eds.), *Multitarget-Multisensor Tracking: Applications and Advances*, Norwood, MA: Artech House, 2000, vol. III, ch. 2.

[20] S. Mori and C-Y. Chong
Effects of unpaired objects and sensor biases on track-to-track association: Problems and solutions.
In *Proceedings of MSS National Symposium on Sensor and Data Fusion*, vol. 1, 2000, 137–151.

[21] S. Mori and C-Y. Chong
Comparison of bias removal algorithms in track-to-track association.
In O. E. Drummond and R. D. Teichgraeber (Eds.), *Proceedings of SPIE*, vol. 6699, Signal and Data Processing of Small Targets, 2007.

[22] G. L. Nemhauser and L. A. Wolsey
Integer and Combinatorial Optimization.
New York: Wiley, 1988.

[23] D. J. Papageorgiou
Track-to-track association using pure association likelihoods.
Raytheon Technical Report, Aug. 2008.

[24] D. J. Papageorgiou and M. Holender
Track-to-track association and ambiguity management in the presence of sensor bias.
In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, WA, July 2009.

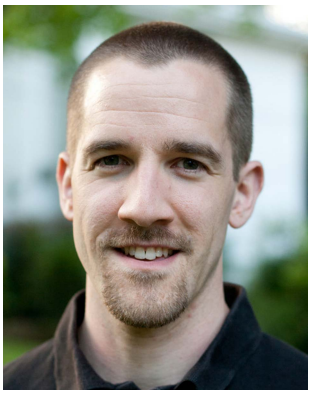
[25] D. J. Papageorgiou and J-D. Sergi
Simultaneous track-to-track association and bias removal using multistart local search.
In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT: IEEE, Mar. 2008.

[26] H. Sherali and W. Adams
A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems.
SIAM Journal on Discrete Math, **3**, 3 (1990), 411–430.

[27] L. D. Stone, M. L. Williams, and T. M. Tran
Track-to-track association and bias removal.
In O. E. Drummond (Ed.), *Proceedings of SPIE*, vol. 4728, Signal and Data Processing of Small Targets, Aug. 2002, 315–329.

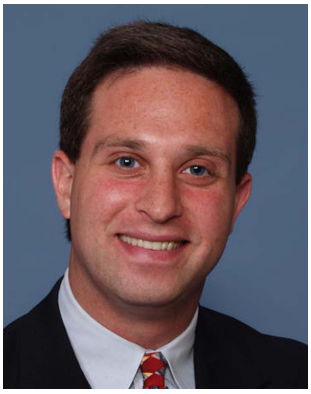
[28] B. Zhou and N. K. Bose
An efficient algorithm for data association in multitarget tracking.
IEEE Transactions on Aerospace and Electronic Systems, **31**, 1 (1995), 458–468.

[29] Y. Zhu
Recent advances and challenges in quadratic assignment and related problems.
Ph.D. dissertation, University of Pennsylvania, 2007.



Dimitri Papageorgiou holds a B.S. in mathematics from the University of North Carolina at Chapel Hill and an M.S. in operations research and industrial engineering from the University of Texas at Austin. He is currently finishing his Ph.D. in operations research under the aegis of George Nemhauser in the H. Milton Stewart School of Industrial and Systems Engineering at the Georgia Institute of Technology.

He is a recipient of a U.S. National Science Foundation Graduate Research Fellowship and was the only person in the country pursuing a doctorate in the field of operations research to receive this honor in his year. As a former member of Raytheon's Center for Discrimination, he has been involved in projects related to sensor resource management, multi-target tracking, and data association, and has co-authored several papers in these areas.



Michael Holender holds a B.S. in mathematics and statistics from Miami University in Oxford, OH as well as an M.S. in industrial and systems engineering from SUNY at Buffalo. He earned his Ph.D. in industrial and systems engineering (operations research) from the State University of New York (SUNY) at Buffalo while studying information fusion.

He is currently a senior systems engineer with the Integrated Defense Systems Division of The Raytheon Company where he has been involved in multi-target tracking and discrimination algorithms. He helped design a new set of algorithms for simultaneous association and classification of objects rooted in the theory of conceptual spaces.

Decision-Level Fusion Performance Improvement From Enhanced HRR Radar Clutter Suppression

BART KAHLER
ERIK BLASCH

Airborne radar tracking in moving ground vehicle scenarios is impacted by sensor, target, and environmental dynamics. Moving targets can be characterized by 1-D High Range Resolution (HRR) Radar profiles with sufficient Signal-to-Noise Ratio (SNR). The amplitude feature information for each range bin of the HRR profile is used to discern one target from another to help maintain track or to identify a vehicle. Typical radar clutter suppression algorithms developed for processing moving ground target data not only remove the surrounding clutter, but a portion of the target signature. Enhanced clutter suppression can be achieved using a Multi-channel Signal Subspace (MSS) algorithm, which preserves target features. In this paper, we (1) exploit extra feature information from enhanced clutter suppression for Automatic Target Recognition (ATR), (2) present a Decision-Level Fusion (DLF) gain comparison using Displaced Phase Center Antenna (DPCA) and MSS clutter suppressed HRR data; and (3) develop a confusion-matrix identity fusion result for Simultaneous Tracking and Identification (STID). The results show that more channels for MSS increase identification over DPCA, result in a slightly noisier clutter suppressed image, and preserve more target features after clutter cancellation. The paper contributions include extending a two-channel MSS clutter cancellation technique to three channels, verifying the MSS is superior to the DPCA technique for target identification, and a comparison of these techniques in a novel multi-look confusion matrix decision-level fusion process.

Manuscript received April 16, 2009, Nov. 24, 2009, April 22, 2010, April 28, 2011; released for publication May 10, 2011.

Refereeing of this contribution was handled by Dr. Robert Lynch.

Authors' addresses: B. Kahler, SAIC, Dayton, OH; E. Blasch, AFRL, Dayton, OH.

1. INTRODUCTION

Many surveillance systems incorporate High Range Resolution (HRR) radar and Synthetic Aperture Radar (SAR) modes to be able to capture moving and stationary targets. Feature-, signature-, and categorical-aided tracking and Automatic Target Recognition (ATR) applications benefit from HRR radar processing. Successful Simultaneous Tracking and Identification (STID) [6, 12, 65] applications exploit feature information to determine the target type and dynamics. Throughout the paper, we use identification, as opposed to recognition, to clarify the process of distinguishing between two targets of the same classification label or allegiance type.

To maximize a search area, airborne systems operate at standoff ranges to detect targets and initiate tracks [3, 5]. After target acquisition and track initiation, tracking systems then transition into a track maintenance mode. However, closely spaced targets require feature analysis to identify the targets. In track maintenance, HRR radar affords dynamic processing analysis for vehicle tracking and signal feature extraction (range, angle, aspect, and peak amplitudes) for target detection and identification [7].

Pattern recognition algorithms applied to ATR problems are typically trained on a group of desired objects in a library to gain a statistical representation of each objects' features. One-dimensional (1-D) HRR classifiers exploit the location and peak amplitude information contained in the HRR signatures [19, 38]. HRR classifiers align input signatures to the library templates or models [16] and determine the best correlation value for the aligned features. HRR ATR algorithms often apply a threshold to the best score to reject questionable objects before making identification or class label decisions. As per the previous work on target identification from HRR signatures, we improve existing capabilities by increasing the peak amplitudes and refine range-bin locations through clutter suppression techniques.

A number of papers have been published that evaluate one-dimensional (1-D) HRR ATR solutions [22, 27, 46, 62, 63]. Classifiers have been developed for correlation [34], Bayes and Dempster Shafer information fusion [11], entropy and information theory analysis [8], and neuro-fuzzy methods [10]. The classifier results have been used for tracking [9, 66] and multi-look HRR identification [53, 67]. Other approaches include eigenvalue template matching [51], Eigen-Extended Maximum Average Correlation (EEMACH) filters [31] and likelihood methods accounting for Rician, amplitude, specular, and diffuse, Cisoid scattering [18]. Since we utilize a combination of sensor and exploitation algorithms (with reported decisions) we are not afforded feature or signal-level fusion options. Using inspiration from the above ATR fusion methods, we incorporate maximum-likelihood Bayesian methods into our novel Confusion Matrix Decision-Level Fusion (CM-DLF) algorithm.

Although the ATR process seems straightforward, misidentification or rejection of an input object as a viable target occurs because of feature extraction differences over different operating conditions. Extended operating conditions (EOCs) such as the target being obscured from the sensor, targets adjacent to another object, and target transitions from moving to stationary and back to a moving state in a traffic scenario unexpectedly alter the features. The importance and impact of EOCs is critical to ATR performance [25]. The quality of the information used in joint tracking, classification, and identification [1, 32, 36, 64] can be determined through Bayes, Dempster-Shafer, or Dezert-Smarandache Theory (DSmT) analysis methods [60]. The clutter-suppressed CM-DLF approach enhances both EOC target identification through minimizing residual range and Doppler noise and enhancing track accuracy through pose angle determination with the correct target shape (features in range-Doppler space).

HRR ATR algorithm performance is impacted by the quality of the features available in the 1-D HRR profiles. Missing target features in training data will alter the library templates formed resulting in poorer identification performance. The presence of EOCs will degrade 1-D test signatures and the corresponding classifier performance. Since the signature data used by ATR algorithms is not always pristine, information fusion methods have been developed such as multi-look ATR, decision level fusion (DLF), and feature level fusion (FLF) in an effort to enhance identification performance from HRR radar data. Improved HRR processing prior to 1-D HRR profile formation (i.e., clutter cancellation) should improve the target features available or reveal more target features, resulting in higher quality 1-D signatures and improved ATR performance.

For many decades, researchers have been developing methods for target identification (ID) through HRR analysis either focused on the radar data itself (e.g., clutter suppression) or the target classification (e.g., pattern recognition methods), which lack the ability to deal with high-density closely-spaced moving target IDs. As simultaneous tracking and identification methods are being applied to urban areas, targets are closer together, have maneuvering dynamics, and are of similar shape. To compensate for these needs, we have coordinated the development of (1) MSS clutter suppression enhancements to deal with closely spaced targets, (2) designed a novel confusion-matrix decision-level fusion approach to take sensor-exploitation likelihood results and update target ID tracks, and (3) combined clutter suppression and CM-DLF for enhanced target signature analysis through movements. Conceptually, target identification improves from having more (a) salient features, (b) spatially and temporally refined features in range-Doppler space, and (c) recursively fused features from different perspectives.

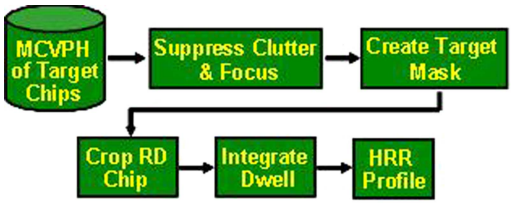


Fig. 1. 1-D HRR profile formation process.

This paper reviews HRR data processing in Section 2, discusses the implementation of a standard two-channel DPCA clutter cancellation method, presents an improved multi-channel signal subspace (MSS) clutter suppression algorithm, and compares the resulting clutter-canceled target chips and target-to-clutter ratios. In Section 3, a multi-look decision level fusion identification method is presented along with performance metrics. Section 4 presents the DPCA and MSS 1-D HRR identification performance in both single-look and multi-look scenarios and Section 5 discusses conclusions and future work.

2. HRR DATA PROCESSING

Focused 1-D HRR radar profiles of *moving targets* may be generated with enhanced target-to-clutter ratios. The moving target is first chipped from the motion-compensated video phase history (MCVPH) radar data for each channel available. The chipped target of the trailing channel is aligned to the target chip of the lead channel for clutter suppression and focusing, as illustrated in Fig. 1. The processing results in a two-dimensional range-Doppler (RD) chip (shown in Fig. 5) that is masked using binary morphology to determine the mean clutter level, target length, and target edges in the chip. The range-Doppler chip is then cropped about the Doppler extent of the target mask before computing the mean of all sub-aperture images. The integration over the dwell time is conducted, which is the duration that the target remains in the radar's beam during each scan. The maximum scatters from each range bin are kept to form the 1-D HRR profile.

Stationary targets from SAR imagery may also be formed into 1-D HRR profiles using a similar process. For targets in SAR imagery, constant-false alarm rate (CFAR) detection is performed first, followed by target mask formation using binary morphology. The formation process crops around the target mask and computes the mean of all sub-aperture images, keeping the maximum scatters from each range bin to form the stationary HRR profile. Shown in Fig. 1 is the general profile formation process flow.

Recent research [19] has shown that HRR profiles formed from SAR imagery of stationary targets have comparable features to profiles of the same target moving at corresponding collection geometries as shown in Fig. 2. The amplitude of the moving target range profile (dashed red line) is lower relative to the stationary target profile (solid blue line) because some of the moving

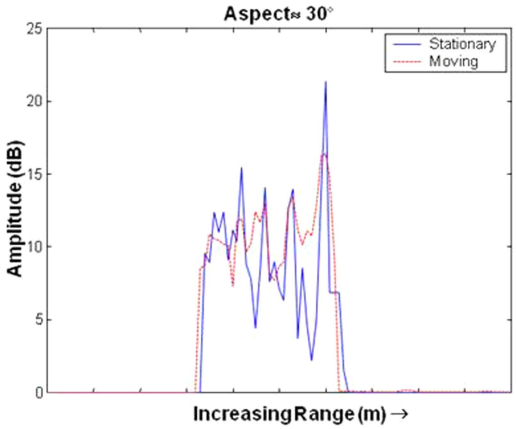


Fig. 2. Comparison of moving/stationary 1-D HRR profiles.

target energy is lost during clutter cancellation while forming the HRR profile. HRR profiles formed from SAR images do not require clutter cancellation in the range profile formation process. However, the strongest scattering features of the target still correlate in range between the moving and stationary target HRR profiles. Since HRR profiles can be formed directly from moving platform HRR collections or extracted from SAR target chips, we are able to utilize collected SAR images for HRR testing and analysis.

2.1. General Clutter Suppression

Clutter suppression of airborne radar data for moving ground targets is a crucial step in target detection and identification. Clutter suppression is needed to enhance the target signature while reducing the competing ground clutter energy surrounding the moving target [13]. As shown in the results in Section 2.4, three visible improvements in Figs. 8–10 that compare two-channel versus three-channel MSS are: (1) better estimate of energy return to range resulting in the estimate of the target length, (2) reduced Doppler clutter to enhance target movement estimates for tracking, and finally, (3) more salient features from which to do a target identification from either template or model matching.

Typically, clutter suppression techniques have the unintentional side effect of reducing some of the target energy while suppressing the ground clutter. Although the target-to-clutter ratio may improve greatly, a reduction in the target features is inevitable, which impacts target tracking and identification performance. The processing of airborne multi-channel radar data to cancel the clutter near moving ground targets can be accomplished through a variety of techniques such as Doppler filtering, space-time adaptive processing (STAP), or displaced phase center antenna (DPCA) processing [13].

Doppler filtering is a technique used with adaptive radars that sense the Doppler distribution of clutter and adjust the radar parameters in an attempt to maximize the signal to clutter ratio. Clutter suppression is accomplished by obtaining a separate coherent output from each channel of an antenna array and applying a unique

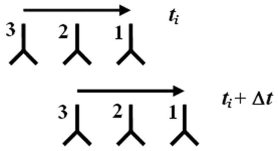


Fig. 3. Three-channel antenna configuration.

complex weight to each channel. Then the weighted channels are added coherently to cancel the clutter energy [13, 21, 33].

A two-dimensional filtering technique known as *Space Time Adaptive Processing* (*STAP*) [21, 28, 29, 49] uses the Doppler frequency, sensor platform velocity, and direction of arrival information to achieve clutter cancellation. Adaptive filter weights are determined for the temporal and spatial domains after sampling a coherent train of pulses. These weights then form a two-dimensional space-time clutter filter that can be applied to the data to eliminate ground clutter. STAP processing is robust to errors and can simultaneously suppress clutter returns and jamming signals [17, 40, 48].

In *DPCA* processing, radar motion compensation reduces the Doppler spread of ground clutter induced by the sensor platform [30, 35, 41]. A multi-channel airborne radar configuration often has a pair of antennas positioned so that as the platform travels in time, the position of the trailing antenna will occupy approximately the same position of the lead antenna at some delta time. Essentially, for a given time interval, one antenna position is fixed. Clutter suppression is accomplished by subtracting the received signal from the trailing antenna at the delta time from the received signal of the lead antenna at the initial time of the processing interval [37, 39, 58].

Both STAP and DPCA are capable of cancelling main beam and side lobe clutter for multi-channel airborne radars with two or more antenna phase centers [40]. In this paper, an available DPCA two-channel algorithm was chosen for comparison to the multi-channel signal subspace algorithm.

A three-channel antenna configuration is shown in Fig. 3, where antenna number 1 is the lead channel for the collected data. The concept of DPCA processing is illustrated in Fig. 3 for a three-channel antenna array configuration. The positions of the antennas are shown at the initial time, t_i, and with platform motion at some time interval, $t_i + \Delta t$, where Δt is the change in time. Through DPCA processing, two antenna positions will appear to be at the same physical location for the array depicted in Fig. 3. Therefore, clutter cancellation is possible where channel 2 at t_i and channel 3 at $t_i + \Delta t$ line up and where channel 1 at t_i and channel 2 at $t_i + \Delta t$ are aligned.

The radar data processed for this paper was collected at X-Band with the aircraft traveling in a linear flight path north of the scene center collecting in spotlight mode at a depression angle of 8.97 degrees and at a weighted resolution of approximately 12 inches. As

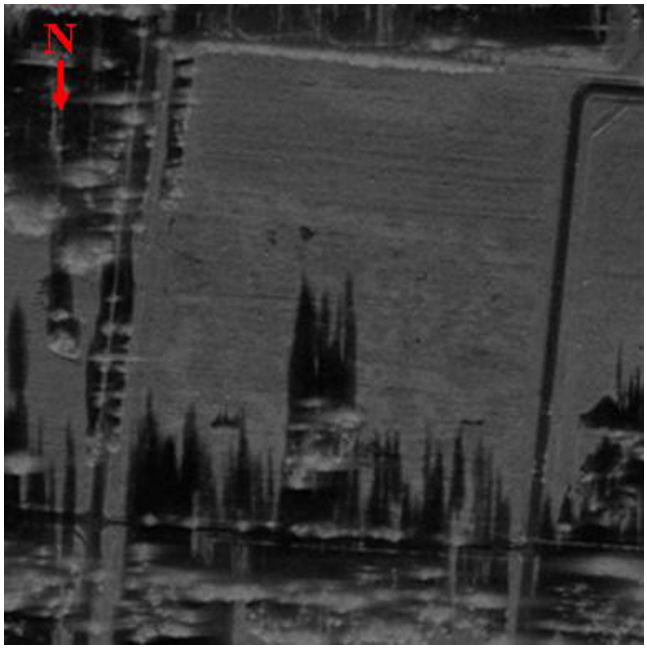


Fig. 4. SAR image of collection site.

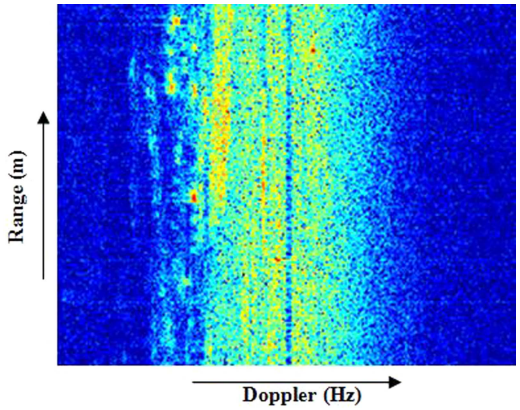


Fig. 5. Example of target chip before clutter suppression.

illustrated in the SAR image of Fig. 4, the center of the collection site was a rectangular grassy area with roads along the western, eastern, and southern borders of the target area. Wooded areas surround the grassy rectangle along the northern, eastern, and southern portions of the scene. In the scenario, civilian vehicles (i.e., trucks and cars) are traveling along the roads in all directions.

The image chips used in the processing discussion are of the test vehicle moving south along the western road. In Fig. 5, an example range-Doppler chip of the target vehicle from channel 1, the lead channel, before clutter suppression is shown. The y-axis is a function of the range bins, which when multiplied by the pixel spacing, is measured in meters (m). The x-axis is in Hz, where the maximum Doppler shift for the clutter is determined as (2∗velocity of the sensor)/wavelength.

The two-channel DPCA processing approach is explained in Section 2.2. Section 2.3 explains the multichannel signal subspace algorithm and the clutter suppression results of the target chip are presented in Section 2.4.

2.2. DPCA Technique

In Section 2.1, the DPCA processing was introduced. The DPCA algorithm used in this work was developed for measured data from a radar array of two antennas oriented along the sensor platform path of travel. In general, the data from the trailing antenna (channel 2 of Fig. 6) is aligned to the lead channel (channel 1 of Fig. 6), where the phases are adjusted so that the aligned channels appear to be at the same location in space, and finally, the channels are subtracted to suppress the stationary clutter. Fig. 6 illustrates the processing steps and data flow of the DPCA technique.

The DPCA algorithm is provided motion-compensated phase history data for both the lead (channel 1) and trailing (channel 2) channels. Channel 2, the trailing channel data, contains extra pulses to address minor offset delays between the channels. *Alignment* of the range and pulse offset is conducted to roughly get channel 2 to approximate channel 1. Then, the antenna patterns are estimated for each channel and an antenna *pattern correction* is applied to channel 2 so that the channels are similar. A *phase correction* is determined in the Doppler compressed domain to account for differences in the frequency direction not already corrected by coarse channel alignment and to address small phase variations between the channels caused by any minor hardware differences in the collection system. The phase correction factor is applied to the data of channel 2. Further *phase adjustments* are determined in the range-Doppler domain and applied to channel 2 to account for any shift in the fast-time samples. A series of additional phase corrections are applied to channel 2 by the DPCA algorithm to improve the *fine alignment* of channel 2 to channel 1 to maximize the target-to-clutter ratio. These

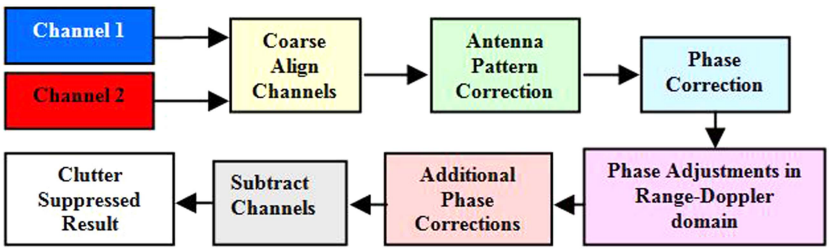


Fig. 6. Two-channel DPCA process flow.

additional corrections require that any target-like objects in the data be avoided so that the target energy is not included in phase correction estimates as was done in the determination of the previous correction factors. The correction factors account for time varying phase differences between both channels being applied to channel 2. Next, a fast-time magnitude and phase correction is applied in the Doppler compressed domain to the data of channel 2. Then a *smoothing technique* is applied to the data resulting in a trailing channel that has been equalized to the lead channel. This completes the alignment process of channel 2 to channel 1. Now that channel 2 appears to be the same as channel 1, the *subtraction* of the channels result in the *cancellation of stationary clutter* in the scene. The baseline clutter suppressed data is represented by $f_d(k,n) = f_1(k,n) - f_2(k,n)$, where f_1 is channel 1, f_2 is the equalized phase history data of channel 2 to that of channel 1, k is the fast-time index, n is the pulse number, and f_d is the clutter-cancelled result.

The DPCA adaptive clutter cancellation method presented will be applied to the data used in this work to ultimately produce the DPCA 1-D HRR profiles. In an effort to improve the 1-D HRR profiles and preserve more target features, a multi-channel signal subspace technique is developed in Section 2.3.

2.3. Multi-channel Signal Subspace Technique

The exploitation of the additional information of a third channel in the phased array radar yields more precise clutter estimation and results in better suppression of unwanted clutter returns. By using the information of all three channels, more target features are preserved in the clutter canceled image [26]. Increasing available target features should translate into better target identification performance. This section will briefly explain the background, the theory behind two channel clutter suppression, and extend this technique to three-channel clutter cancellation.

2.3.1. Signal Subspace Background

The *Multi-channel Signal Subspace* (MSS) technique is based on 2-D adaptive filtering principles. The process has been applied to a wide variety of data processing problems in the literature [54] such as SAR change detection, [47] image fusion of radar surveillance data, [56, 57] medical imaging, and video processing [20, 54]. Most of the work with signal subspace processing has focused on data pairs either separated spatially (e.g., two channel phased array radar data) or separated temporally (e.g., such as electro-optical images collected at different times) as discussed in the literature by Soumekh and others [54, 55, 57].

The development of a true multi-channel, greater than two, signal subspace algorithm for use with a multi-channel radar system consisting of a planar antenna array of 22 receiver channels seemed likely [2]. However, the received power at each channel was too weak to form an image of sufficient quality for further processing. This issue was resolved by splitting the data from the 22 channels into a pair of 11 receiver channel groups that were summed to improve the signal to noise level [57]. Once the planar antenna array is represented by two receive channels, the signal subspace processing technique is applied to clutter-cancel the data. In the next section, the process for two-channel clutter suppression will be explained.

2.3.2. Dual-Channel Signal Subspace Technique

The dual-channel radar system discussed in this section will have a pair of antennas in a phased array similar to what is illustrated in Fig. 3, but without the third channel being present. Channel 1 will be the lead channel and channel 2 will be the trailing channel. In keeping with the convention found in the literature, let $f_1(x,y)$ represent the range-Doppler image formed from the motion-compensated data from channel 1 over a coherent processing interval (CPI) of 128 ms. Then, $f_2(x,y)$ will be the range-Doppler image formed from the motion-compensated data from channel 2 after a slow-time alignment with channel 1. Since the channel 2 range-Doppler image is a linear combination of channel 1 and any shifted versions, $f_2(x,y)$ can then be modeled by [54, 20], $f_2(x,y) = f_1(x,y) \otimes h(x,y) + f_e(x,y)$; where $\otimes$ is the convolution operator, $f_e(x,y)$ represents the target motion in the range-Doppler image, and $h(x,y)$ is the impulse response representing the relative shift in each range-Doppler image due to differences in the two receive channels of the sensor system.

Gain and phase ambiguities caused by known and unknown factors, such as differences between the antenna patterns or antenna vibration, in the two receive channels may dominate the moving target signature in the imagery. These differences are treated as an error signal in the collected data. The DPCA approach reduces the error signal to a set of pre-determined functions that are estimated and accounted for deterministically. The MSS technique applied to a dual antenna sensor system views the error estimation process as completely stochastic.

Signal subspace theory estimates $h(x,y)$ from $f_1(x,y)$ and $f_2(x,y)$ resulting in the error function, $\hat{h}(x,y)$ [54, 56]. This is accomplished by minimizing the squared error between $f_2(x,y)$ and its estimated version given by

$$\hat{f}_2(x,y) = \hat{h}(x,y) \otimes f_1(x,y) \tag{1}$$

where $\hat{f}_2(x,y)$ is determined by projecting $f_2(x,y)$ on to a set of orthogonal basis functions defined by $f_1(x,y)$ [54]. The orthogonal basis functions can be computed using any one of accepted decomposition/orthogonalization techniques such as singular value decomposition, QR orthogonalization, or the Gram-Schmidt procedure. QR orthogonalization was used in the MSS implementation that generated the results of this paper where in practice $\hat{h}(x,y) \otimes f_1(x,y)$ is estimated instead

of $\hat{h}(x,y)$. In general, the spatially-invariant difference over the entire image is represented by [20, 54],

$$\hat{f}_d(x,y) = f_2(x,y) - \hat{f}_2(x,y). \quad (2)$$

To suppress unwanted clutter in radar data, the error function is estimated with overlapping odd-sized blocks over the entire image to account for the spatially varying nature of the phase in the imagery. The entire range-Doppler image is divided into rectangular blocks containing an odd number of pixels and processed to estimate the error function. The blocks of image pixels were moved so that some portion of the rectangular patch overlapped a previously processed block until the entire subdivided image had been processed. This results in a clutter-cancelled image given by [56],

$$\hat{f}_d(x_i,y_i) = \sum_{l=1}^{L} [f_2(x_i,y_i) - \hat{f}_2(x_i,y_i)] I_l(x_i,y_i) \quad (3)$$

for a two channel phased array radar system. L is the number of overlapping blocks processed, i is the odd number of pixels per block, and I_l is an identity matrix. The MSS implementation in this paper used square patches in the processing represented by (x_i,y_i), but in general a rectangular block represented by (x_i,y_j) could be used for an i-by-j-dimensional block. The next section discusses the extension of this technique to data collected with a three-channel phased array radar.

2.3.3. Three Channel Signal Subspace Technique

The two-channel signal subspace method explained in Section 2.3.2 is extended for use with all three channels of the phased array radar depicted in Fig. 3. Once again, the lead channel will be channel 1 and the trailing channels will be 2 and 3. The Multi-channel Signal Subspace (MSS) extension to three channels will first project the range-Doppler image formed from the aligned motion compensated data of channel 2, $f_2(x,y)$, on to the basis functions defined by the range-Doppler image formed from the motion compensated data of channel 1, $f_1(x,y)$, and determine the spatially varying difference, $\hat{f}_{d2}(x_i,y_i)$, given by Equation 3. The resulting range-Doppler difference image of channels 1 and 2 is treated as a *new* independent channel, $f_4(x,y)$, as shown by

$$f_4(x,y) = \hat{f}_{d2}(x,y) = f_2(x,y) - \hat{f}_2(x,y). \quad (4)$$

Then the range-Doppler image formed by the aligned motion compensated data of channel 3, $f_3(x,y)$, is projected on to the basis functions defined by the range-Doppler image formed from the motion compensated data of channel 2, $f_2(x,y)$. The spatially varying difference, $\hat{f}_{d3}(x_i,y_i)$, from (3) is then determined. The resulting range-Doppler difference image of channels 2 and 3 in $f_5(x,y) = \hat{f}_{d3}(x,y) = f_3(x,y) - \hat{f}_3(x,y)$ is treated as a second *new* independent channel, $f_5(x,y)$, at a slightly different look angle.

Now the second *new* independent channel, $f_5(x,y)$, is projected on to the orthogonal basis functions of the first *new* independent channel, $f_4(x,y)$, represented by $\hat{f}_5(x,y) = f_4(x,y) \otimes \hat{h}_{45}(x,y)$.

The three-channel spatially-invariant difference image is represented as $\hat{f}_d(x,y) = f_5(x,y) - \hat{f}_5(x,y)$. The block processing represented by Equation 3 was applied to account for the spatially varying nature of the range-Doppler images.

Since each of the *new* independent channels is essentially a clutter-canceled range-Doppler image, this technique represents the fusion of two dual-channel clutter-suppressed range-Doppler images. The resulting *clutter-suppressed* range-Doppler image should contain more target features from the slightly different viewing antenna geometries in the array. The MSS method improves target features without enhancing any residual clutter in the *new* input images. Examples of this processing are presented in the section that follows.

2.4. Clutter-Suppression Results

The clutter-suppressed range-Doppler chips presented in this section were generated from the same part of the collected data discussed in Section 2.1. The moving target, a sedan, is slowing down while heading south, away from the radars' location. All of the range-Doppler chips presented in this section have a dynamic range of 50 dB with Doppler increasing from the left of the image to the right, and range increasing from the bottom of the image to the top. The DPCA algorithm result is presented first, then the two-channel MSS processed chips, and finally the three-channel clutter-suppressed result. The signal-to-noise ratio for all of the clutter-suppressed range-Doppler chips is computed for algorithm performance comparison.

The implementation of the DPCA algorithm required the first channel to be the lead channel and limited the amount of shifting that may occur to align the two channels. Therefore, only channels 1 and 2 could be processed to yield a clutter-cancelled range-Doppler image. The result is shown in Fig. 7. As stated earlier, the dynamic range is constant for all the results present in this section. However, adjusting the dynamic range of the DPCA range-Doppler chip would help better define the target.

Although the DPCA method could only produce clutter-cancelled chips from two of the three channels available, the multi-channel signal subspace (MSS) technique utilized all three channels in the processing. Fig. 8 is the clutter-suppressed range-Doppler image produced from channels 1 and 2. In comparing Fig. 8 to Fig. 7, the MSS approach does a better job of clutter cancellation than the baseline technique using the same data channels.

In Fig. 9, the clutter cancelled result of the MSS algorithm using channels 2 and 3 is presented. The relative reduction of clutter is similar to that of Fig. 8.

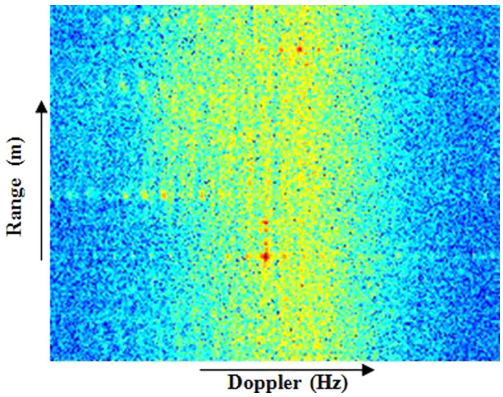


Fig. 7. DPCA clutter suppression technique.

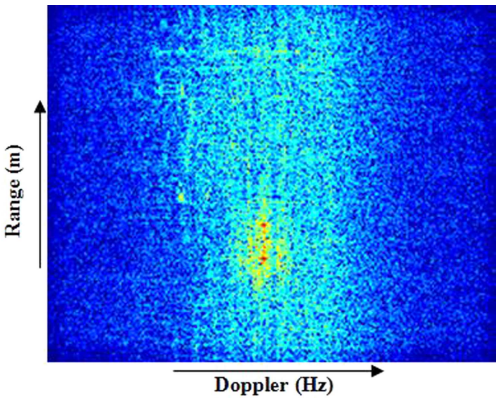


Fig. 8. MSS Two-channel clutter suppression technique: channels 1 and 2.

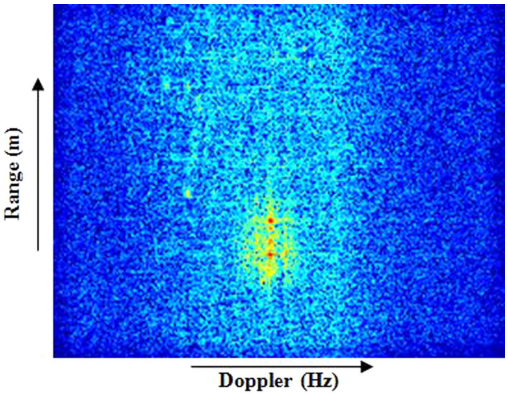


Fig. 9. MSS Two-channel clutter suppression technique: channels 2 and 3.

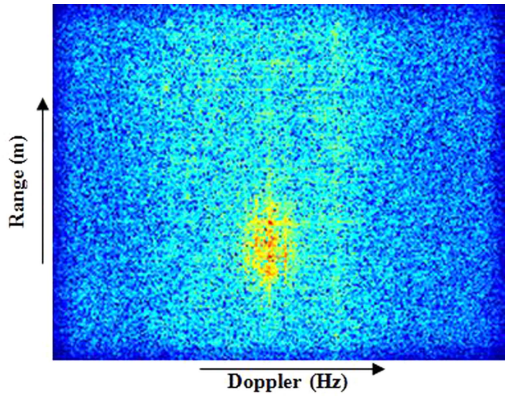


Fig. 10. MSS Three-channel clutter suppression technique.

Close examination of Figs. 8 and 9 reveal scattering from different locations of the target as well as more features in the range-Doppler imagery. This is caused in part by minor variations in the collection geometry due to the spacing of the antennas in the phased array radar.

Found in Fig. 10 is the clutter suppressed range-Doppler chip produced by the enhanced MSS algorithm using all three channels of the motion compensated data. A minor reduction in the level of clutter cancellation can be seen when comparing the results of Fig. 10 to that of Figs. 8 and 9. However, careful examination of the range-Doppler image in Fig. 10 shows more target features are present. The three-channel clutter suppressed image has a signal to noise level comparable to that of the MSS two-channel clutter-cancelled results and is an improvement over the baseline technique.

Finally, a signal-to-clutter ratio was determined for the chips presented in this section to help gauge the relative performance levels of the various techniques. This ratio was determined by finding the largest pixel value in the image; which is the brightest point on the target and dividing it by the average clutter in a one pixel wide frame around the entire range-Doppler chip. A comparison of the signal-to-clutter levels for the range-Doppler images formed from the three techniques discussed in this paper can be found in Table I.

TABLE I
DPCA vs. MSS Target to Clutter Ratio Comparison

DPCA processing	33.20 dB
MSS: 2 channel SS: channels 1 and 2	42.81 dB
MSS: 2 channel SS: channels 2 and 3	42.99 dB
MSS: 3 channel SS	42.57 dB

The MSS performance scales based on the comparable target-to-clutter ratios for both the two-channel and three-channel processing, with the three-channel MSS method resulting in a slightly noisier clutter-suppressed image, but with the added benefit of more target features being preserved after clutter cancellation. The results in Table I indicate that the MSS technique for traditional two-channel clutter cancellation and for multi-channel clutter suppression performs much better, nearly 10 dB in target to clutter ratio, than the DPCA method.

3. TARGET IDENTIFICATION FUSION

The ability to perform track and identity fusion requires sensor-processed classifications/identifications from different levels. Multi-target data association algorithms that accurately track targets in the presence of clutter assume that the detected targets can be tracked from a sequence of center-of-gravity and pose positional data. Detected classification can effectively discern the

target for a given scenario using experience of target movement, training, or predicted information. For example, two targets of the same type may be crossing in space, but since they can not occupy the same location, they would each have a different orientation relative to a sensor. By exploiting the orientation, velocity, and multi-resolution feature information, each target can be assessed for the correct track-ID association.

The capability of a sensor to track and identify targets simultaneously requires the target center, the target pose, and neighboring characteristics to discern salient features for target type association. For example, features [59] can be used to identify targets with a level of uncertainty. However, if many features are fused, the identity improves and helps eliminate clutter. The tracker must use the available features to discern an object (*identify* a target), which is a subset of Automatic Target Recognition (ATR). Certain features are inherently more useful in recognizing a target than others, but obtaining these features is a function of sensor resolution and collection geometry.

The 1-D HRR ATR known as the Baseline Automated Recognition of Targets (BART) algorithm has been used to generate identification results [22, 27]. BART is a template matching algorithm using the dominant range-space eigenvector. Eigen-templates have been used for 2D ATR problems using electro-optical [43], SAR, [42, 43], and Forward-Looking IR (FLIR) analysis [14, 15]. In each of these methods, the eigen-template matching provides a stable analysis for a single look. The eigenvector approach was then adapted and refined by Shaw [4, 52, 45] and others [18] for 1-D template formation using HRR profile data.

By leveraging knowledge about target features, fusion algorithms can significantly reduce processing time for tracking and identifying targets. For separated targets, resources may exist to identify each target. However, when resources and processing time are limited, a trade-off exists between the identification and tracking of a target. In the case of multiple ATR systems observing the same area, the HRR profiles can be at significantly different sensor-to-target geometries. Different geometries result in different features for target classification. In such a case, a decision-level fusion approach is a good solution since the ATR decisions are fused and not the features of the target signatures. By leveraging knowledge about target types, fusion algorithms can significantly reduce processing time for tracking and identifying targets. Increased robustness is achieved with a multi-look approach utilizing the eigen-template feature analysis [22], summarized by a classifier confusion matrix, and combined for enhanced HRR target identification.

3.1. Decision Level Fusion (DLF) Method

The decisions from an ATR are often stored in a confusion matrix (CM), which is an estimate of likelihoods. For the single-look ATR performance, these estimates are treated as priors [61]. Decisions from multiple ATRs or from multiple looks from different geometric perspectives are fused using the Decision Level Fusion (DLF) technique presented. With respect to the DLF, the CMs represent the historical performances of the ATR system. Assume that we have two ATRs each described in a confusion matrix designated as C^A and C^B. The elements of a confusion matrix are $c_{ij} = \Pr\{\text{ATR decides } o_j \text{ when } o_i \text{ is true}\}$, where i is the true object class, j is the assigned object class, and $i = 1,\ldots,N$ for N true classes. The CM elements can be represented as probabilities as $c_{ij} = \Pr\{z = j \mid o_i\} = p\{z_j \mid o_i\}$. To determine an object declaration, we need to use Bayes' rule to obtain $p\{o_i \mid z_j\}$ which requires the class priors, $p\{o_i\}$. We denote the priors and likelihoods as column vectors[1]

$$p(\bar{o}) = \begin{bmatrix} p(o_1) \\ p(o_2) \\ \vdots \\ p(o_N) \end{bmatrix} \qquad p(zj \mid \bar{o}) = \begin{bmatrix} p(zj \mid o_1) \\ p(zj \mid o_2) \\ \vdots \\ p(zj \mid o_N) \end{bmatrix}. \tag{5}$$

For M decisions, a confusion matrix would be of the form

$$C = \begin{bmatrix} p(z1 \mid o_1) & p(z2 \mid o_1) & \cdots & p(zM \mid o_1) \\ p(z1 \mid o_2) & p(z2 \mid o_2) & \cdots & p(zM \mid o_2) \\ \cdots & \cdots & \ddots & \cdots \\ p(z1 \mid o_N) & p(z2 \mid o_N) & \cdots & p(zM \mid o_N) \end{bmatrix}. \tag{6}$$

The joint likelihoods are similar column vectors, where we assume independence for two confusion matrices A and B (denoted here as superscripts),

$$p(z_j^A, z_k^B \mid \bar{o}) = \begin{bmatrix} p(z_j^A \mid o_1) \cdot p(z_k^B \mid o_1) \\ p(z_j^A \mid o_2) \cdot p(z_k^B \mid o_2) \\ \cdots \\ p(z_j^A \mid o_N) \cdot p(z_k^B \mid o_N) \end{bmatrix} \tag{7}$$

where k is used to distinguish between the different assigned object classes between the two confusion matrices when the CMs are not symmetric.

Using the priors and the likelihoods, we can calculate *a posteriori* from Bayes' Rule

$$p(\bar{o} \mid z_j^A, z_k^B) = \frac{p(z_j^A, z_k^B \mid \bar{o}) p(\bar{o})}{\sum_{i=1}^{N} p(z_j^A, z_k^B \mid \bar{o}) p(\bar{o})}. \tag{8}$$

[1] Based on FITE Memo, 15 May 2006, from Tim Ross.

```
function [d, pObarZaZb] = fuseCMdecisions(za, zb, Obar)
        CA = getConfusionMatrix(1);
        CB = getConfusionMatrix(2);
        pZaObar = CA(:,za);
        pZbObar = CB(:,zb);
        pZaZbObar = pZaObar .* pZbObar;
        posteriorNum = pZaZbObar .* pObar;
        posteriorDen = sum(posteriorNum);
        pObarZaZb = posteriorNum / posteriorDen;
        [junk, d] = max(pObarZaZb);
        return
```

Fig. 11. Confusion matrix pseudo code.

Note that there are similar column matrices for the posteriors $p(\bar{o} \mid zj)$ and $p(\bar{o} \mid z_j^A, z_k^B)$. A decision is made using the maximum likelihood estimate

$$d_i = \arg\max_{j;k} p(o_i | z_j^A, z_k^B) \tag{9}$$

where the final decision of the true object class i is determined from the largest value from the vector.

Note that the subscripts indicate the value of a variable and the superscripts indicate the ATR source. For example, $z^A = z_3$ indicates that source A made a decision z_3; where source A might be the first look of a HRR ATR and decision z_3 might be target type "sedan." The absence of a superscript implies an unspecified single source. We represent the particular states from each source with the subscripts a and b such as $z^A = z_a^A$ indicating that source A's decision was z_a.

For the developments of the pseudo code, shown below in Fig. 11, we shorten the notation to $z^A = z_a$, while keeping track of the confusion matrix source A or B.

3.2. Naïve Bayes DLF Pseudocode

Inputs to the fuser are the decisions of ATR A and B, i.e., z_a and z_b respectively. The output of the fuser is the decision d based on a maximum a posteriori probability (MAP) decision rule, where the posterior is $p(\bar{o} \mid za, zb)$. The fuser must know the prior probabilities $p(\bar{o})$ and the confusion matrices (one for each source).

Pseudo code for DFL is represented as:

- za = z_a and zb = z_b are the integer decisions between $1 \ldots M$ of sources A and B, respectively
- pObar = $p(\bar{o})$ is a vector of priors, represented as either constants or input variable
- CA = C^A and CB = C^B are the confusion matrices derived from sources A and B, respectively
- pZaObar = $p(z_a \mid \bar{o})$ and pZbObar = $p(z_b \mid \bar{o})$ are the likelihoods as extracted columns from the confusion matrices [pZaObar = CA(:, za); and pZbObar = CB(:, zb)]
- pZaZbMbar = $p(z_a, z_b \mid \bar{o})$ is the joint likelihood derived from the point-wise product of the source likelihoods (pZaZbObar = pZaObar.∗pZbObar);
- pObarZaZb = $p(\bar{o} \mid z_a, z_b) = (p(z_a, z_b \mid \bar{o})p(\bar{o}))/(\sum_{i=1}^{N} \cdot p(z_a, z_b \mid \bar{o})p(\bar{o}))$
  —the numerator is:
  posteriorNum = pZaZbObar.∗pObar;
  —the denominator is:
  posteriorDen = sum(posteriorNum);
  —pMbarZaZb = posteriorNum/posteriorDen;
- d = max(pObarZaZb), which is the fused decision, $d_i \ni p(o_i \mid z_a, z_b) \geq p(o_i \mid z_a, z_b) \quad \forall i, j$ where $i, j \in 1, \ldots, N$.

The DLF function pseudo code is presented for verification.

3.3. Metric Presentation

We used the eigen-value HRR target identification approach as a baseline method [27]. The likelihood vectors were compiled into a confusion matrix (CM). Thus, each single look provided a full analysis of the classifier, presented as a CM, for all target comparisons. The likelihood vectors of the confusion matrix allowed for a more thorough analysis with such performance criteria as declaration, P_D, and false alarm, P_{FA}, probabilities. The confusion matrix lists a set of likelihood values with the real targets as the rows $\{T_1, \ldots, T_N\}$, and the testable hypothesis as the columns $\{T_1, \ldots, T_N, \text{other}\}$. For example, if the true target is T_1, the CM is

$$CM = \begin{array}{c} \\ T_1 \\ T_2 \\ \\ T_N \end{array} \begin{array}{c} \begin{array}{ccccc} T_1 & T_2 & & T_N & \text{Other} \end{array} \\ \begin{bmatrix} A & B & \cdots & B & O \\ E & D & \cdots & D & O \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ E & D & \cdots & D & O \end{bmatrix} \end{array}. \tag{10}$$

Selecting the likelihood values in the CM, one can compare the performance metrics for different size CMs. From the CM and a defined target-to-confusion ratio as m (as set by the operational ATR requirements), a set of metrics can be identified to support analysis including

$$P_{\text{Declaration}} = \frac{A}{A + B} \tag{11}$$

$$P_{\text{FalseAlarm}} = \frac{E}{E + D} \tag{12}$$

$$P_{\text{CorrectClassification}} = \frac{m \cdot PD}{(m \cdot P_D) + P_{FA}}. \tag{13}$$

Using Fig. 21 as an example, let A be the fact that target 1 is choice (row) and that target 1 is declared (column) by the ATR to produce a normalized likelihood of 0.63. The rows are normalized to one but rounding errors lead to values close to but not exactly equal to 1. $B = 0.075 + 0.12 + 0.56 + 0.042 = 0.293$. $E = 0.039 + 0.047 + 0.081 + 0.12 + 0.21 = 0.497$. O is the entire right column of $0.084 + 0.02 + 0.054 + 0.039 + 0.025 + 0.21 = 0.432$. Finally, D is the remaining value $D = 4.146$. Using the results from Fig. 21, then $P_{\text{Dec}} = 0.63/[0.63 + 0.293] = 0.683$. $P_{FA} = 0.497/[0.497 + 4.146] = 0.107$. If we let $m = 1$, then $P_{CC} = 0.683/[0.683 + 0.107] = 0.865$. These metrics are important in the fact that the

system is not forced to make a target declaration (column other) as well as ability to discern whether there is enough evidence to declare a target in the library [50].

By choosing a different target truth (row) to selection (column) as defined by the diagonal, the process is repeated for each true target and the resulting metrics are summed and normalized for the number of targets. For example, the second true target is

$$CM = \begin{array}{c} \\ T_1 \\ T_2 \\ \vdots \\ T_N \end{array} \begin{array}{c} \begin{array}{cccccc} T_1 & T_2 & & \cdots & T_N & \text{Other} \end{array} \\ \begin{bmatrix} D & E & D & \cdots & D & O \\ B & A & B & \cdots & B & O \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ D & E & D & \cdots & D & O \end{bmatrix} \end{array}. \tag{14}$$

4. IDENTIFICATION PERFORMANCE

Identification (ID) performance varies over three operating conditions: environment, sensors, and targets. To demonstrate environment variations, we compare adjacent versus separated target IDs for moving targets in Sections 4.1 and 4.2. To highlight the variations in sensors, we acknowledge the variations between the sensor-processing methods (DCPA versus MSS) throughout Section 4. Finally, for target variation, we show the confusion matrices results for single-look and multi-look comparisons in Sections 4.3–4.5.

The results that follow are from collected HRR profiles from moving targets. Simultaneous target tracking and ID requires using HRR radar mode that supports feature analysis in-between point movements (Moving Target Indicator—MTI mode) and 2-D stationary images (Synthetic Aperture Radar—SAR mode). In addition to enhanced target ID through higher-diagonal CM results, the clutter-suppression results demonstrate improved target localization. HRR feature analysis begins with aligning the HRR profile. Higher signature matches presented in the CM cell's results are indicative of more feature matches (including the length of the HRR target profiles).

Single-look confusion matrices were produced for 1-D HRR profiles formed from DPCA and MSS clutter canceled target chips for ten ground vehicles traveling along the roads shown in Fig. 4. Obscuration from nearby vegetation along the streets impacted identification performance depending on collection geometry. The DLF technique was then applied to five DPCA and five MSS single-look confusion matrices each produced with a unique sample set. The results of these experiments are presented in the subsections that follow.

4.1. Adjacent Vehicles

A subset of the data was selected to address a common target tracking issue, vehicle adjacency, and to compare identification performance among the various types of clutter canceled data in benign conditions. Three cars moving south along the western road shown in Fig. 4 were chosen because the vehicles are in the open and not obscured by vegetation. A bend in the road required the vehicles to decelerate and cluster closer together (less than one car length apart) while making the turn. The vehicle data near the bend in the road was divided from the vehicle data of the cars traveling south toward the bend in the road, creating two data sets: adjacent and separated. Note that these results are for an aspect angle centered around 180 degrees or near the rear of the vehicles.

Confusion matrix results were produced for the closely associated vehicles near the turn in the road. As targets slowed approaching the turn, the data was treated as the adjacency data set. The lead target had the fewest samples, and the trailing target had the most samples since it was waiting for the other targets to make the turn. Using the ID counts for each sample, we turned them into the scores in the single-look CM. Identification performance for the first two vehicles to arrive at the bend in the road was consistently poorer than the trailing vehicle that was less obscured. Note that the data included in Section 4.1 was used later in the overall results.

The identification performance using data sets for all clutter suppression methods are found in Figs. 12–15.

4.2. Separated Vehicles

The data of the three vehicles separated while traveling in the open was used to produce a "best case" identification performance comparison of each clutter suppression method. The recognition results for separated targets are presented in Figs. 16–19. It is noted from the CM, that the ID performance of all vehicles increases from ~ 0.5 to ~ 0.85. The ID performance of separated targets can be used in target confirmation, track maintenance, as well as afford resources to be applied to other search areas to acquire targets. To compare separated versus adjacent methods using the clutter cancellation, we summarize the results from the CMs.

A mean identification performance was computed for each matrix by averaging the diagonal of the CMs and is presented in Fig. 20 for comparison of the relative identification performance gains achieved by the techniques used. The three-channel signal subspace showed the best overall performance, followed by two channel signal subspace clutter cancelation with the two-channel DPCA having the poorest performance of the techniques compared. As expected, a high level of target identification of separated vehicles in benign conditions was achieved for all techniques examined.

4.3. Single-Look Performance

The identification performance comparisons presented in Sections 4.3, 4.4, and 4.5 are between the best (three-Channel MSS) and worst (DPCA) performing techniques in Sections 4.1 and 4.2.

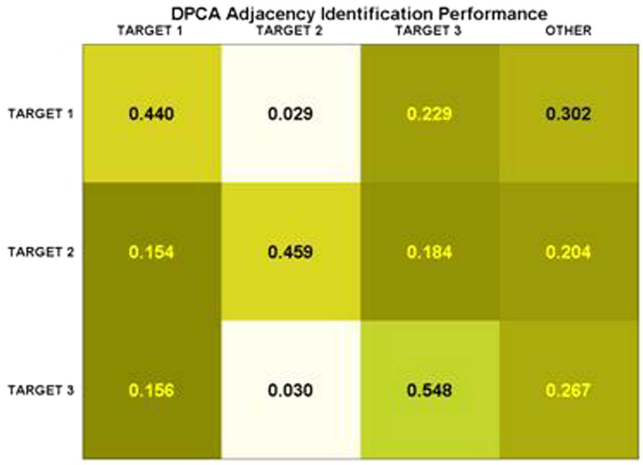


Fig. 12. DPCA single-look performance with target adjacency.

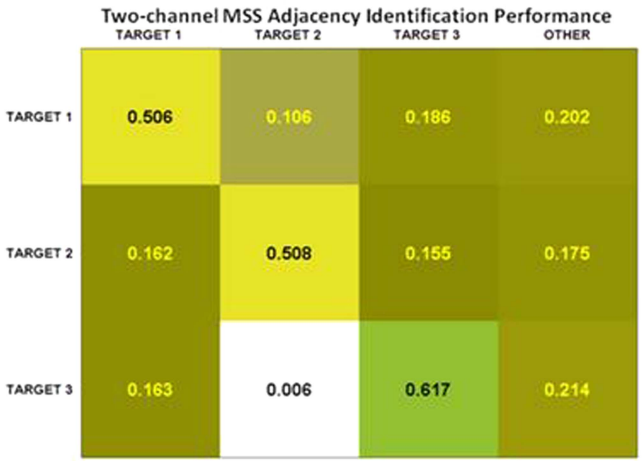


Fig. 13. Two-channel (1 and 2) MSS single-look performance with target adjacency.

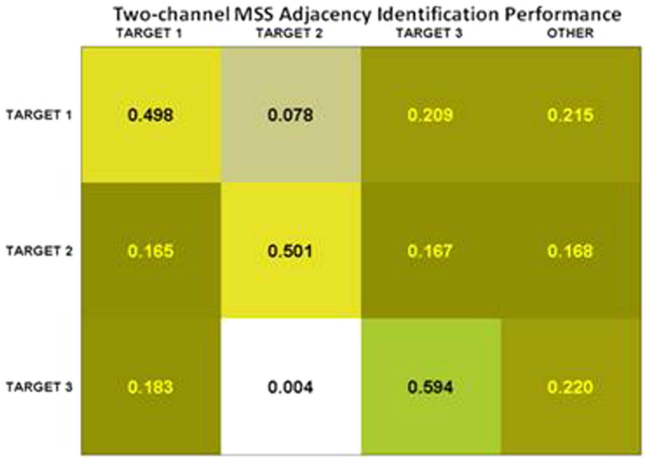


Fig. 14. Two-channel (2 and 3) MSS single-look performance with target adjacency.

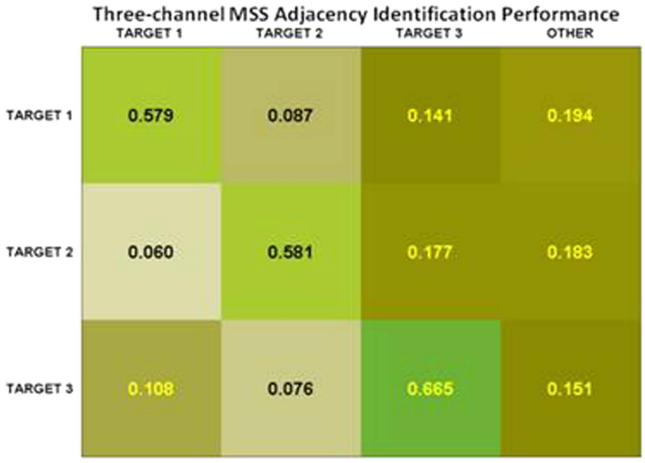


Fig. 15. Three-channel MSS single-look performance with target adjacency.

Five vehicles were selected for the library and the remaining five vehicles were used as confusers as was done in the companion papers [23, 24] for a total of ten templates. The libraries of target profiles were created from the off-line collected target signatures for this data set and compared to the on-line test data as used in [6]. To ensure 360 degree coverage, we used the entire data set to locate enough signatures to develop a 360 degree target signature database for training. After capturing the necessary training set, we used the remaining data for testing. For example, of the 1800 samples, we used about 1500 for training and the remaining 300 for testing.

The DPCA single-look identification results are shown in Fig. 21 with a mean target identification rate of 65%. The three-channel MSS single-look 1-D HRR ID performance is presented in Fig. 22 with an improved mean identification rate of 73.6% relative to the DPCA results.

The distribution of the confuser vehicles was spread across the not-in-library row for both the DPCA and MSS confusion matrices indicating no strong bias toward a library object.

A comparison of the receiver operator curve (ROC) associated with each of the HRR clutter cancelled data sets is presented in Fig. 23. The MSS clutter-suppressed data performs better with respect to the DPCA processed data.

4.4. Multi-look DLF Performance

Wider angle changes (different perspectives) would increase the ATR results. Instead of processing wider (n-channel) clutter suppression, we utilized Decision-Level Fusion (DLF) to incorporate angle changes for enhanced ATR results. The test data used for the DLF was the same as that used to create the single-look analysis. The DLF data included the entire scenario of the ten targets (five selected, five confusers) moving through different operating conditions of adjacency and separation, while the single-look analysis was a subset of only three targets in the specified operating condition over a shorter time interval.

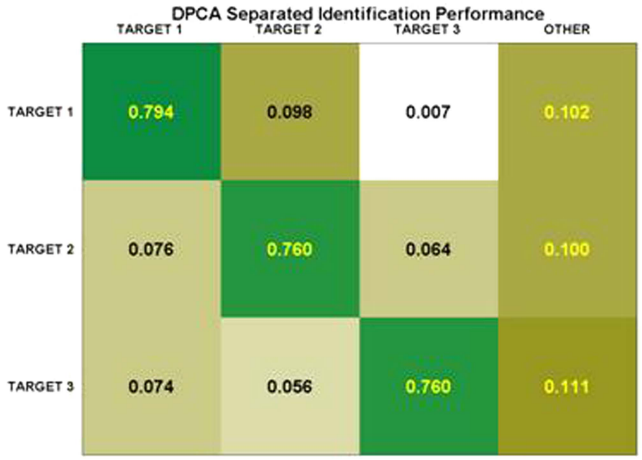


Fig. 16. DPCA single-look performance with separated targets.

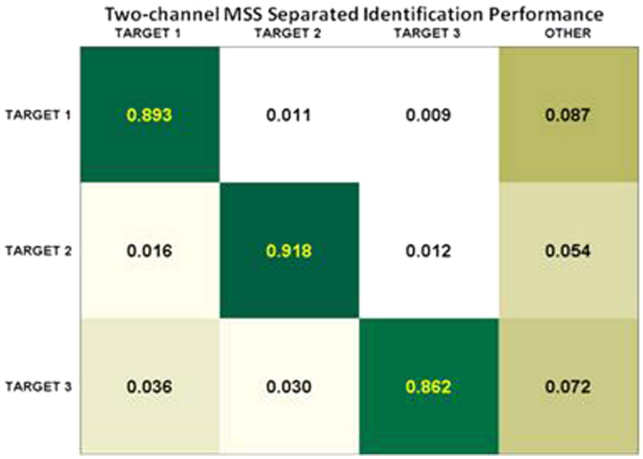


Fig. 17. Two-channel (1 and 2) MSS single-look performance with separated targets.

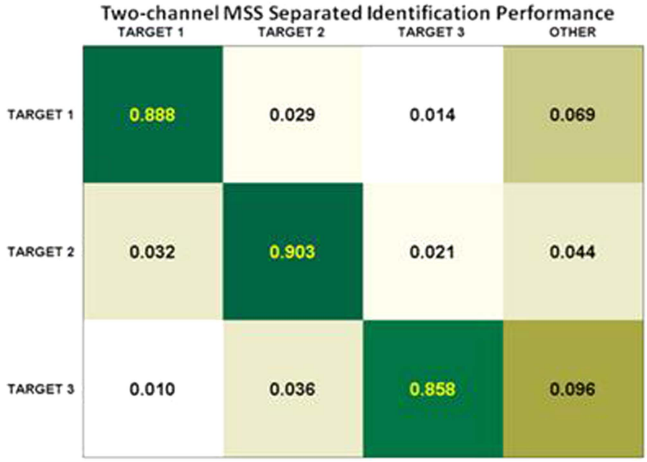


Fig. 18. Two-Channel (2 and 3) MSS single-look performance with separated targets.

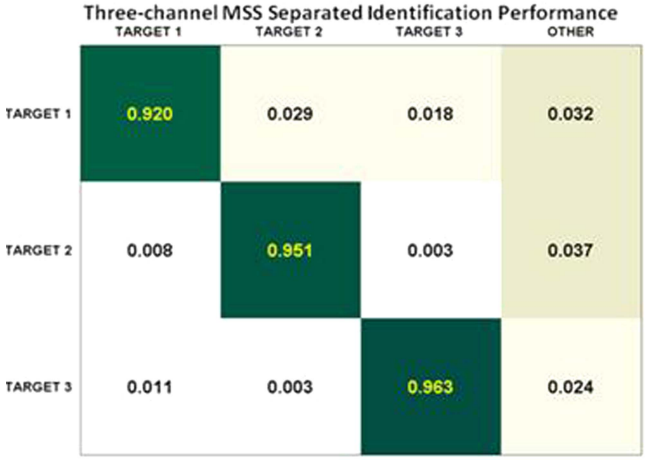


Fig. 19. Three-channel MSS single-look performance with separated targets.

The DLF algorithm described in Section 3.2 was run with confusion matrices 1 and 2 with the remaining confusion matrices being incorporated sequentially and the fused decision from the previous run being treated as prior knowledge of the targets of interest. This procedure was followed for both the DPCA and MSS single-look matrices. The fused DPCA target recognition results can be seen in Fig. 24 with significant performance gains relative to the single-look DPCA results of Fig. 21 and improved performance relative to the MSS single-look results of Fig. 22. Since the multi-look DPCA is better than the single-look MSS, there is value in utilizing multi-look DLF no matter which clutter suppression technique is used.

The best identification results for this scenario are shown in the MSS confusion matrix of Fig. 25 for multi-look DLF. The DLF off-diagonal target confusion was significantly reduced while correct ID was greatly enhanced relative to the DPCA processed data or single-look MSS target recognition. Average MSS DLF target ID increased to 89.2% for this moving target scenario relative to the single-look average DPCA vehicle recognition performance of 64.8%. Since the targets in the dynamic scenario are not always well separated, through DLF and three-channel MSS clutter suppression, the average target ID was better than the average DCPA best single look condition (Fig. 16) and equivalent to that of the average two-channel MSS best single look condition (Figs. 17 and 18).

A ROC comparison of the multi-look DLF performance is shown in Fig. 26. The Multi-channel Signal Subspace clutter suppressed data set has the best performance with respect to the fused DPCA data set and the single-look target identification results. The single-look performance of the MSS data set is comparable to the fused DPCA performance as illustrated by both the confusion matrices of Figs. 22 and 24 and the ROC curves found in Figs. 23 and 26. To further assess the similarities and differences in target identification performance relative to enhanced clutter suppression and fusion technique, the metrics of Section 3.2 are used on a per target basis and presented in the section that follows.

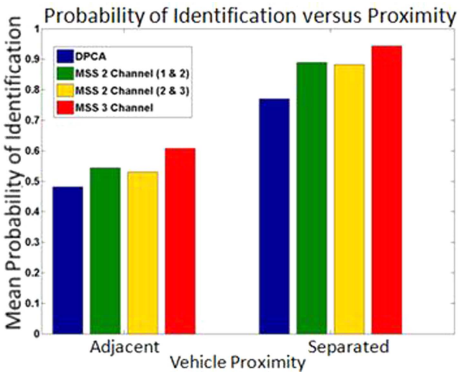


Fig. 20. Comparison of Mean Single-Look Identification Performance.

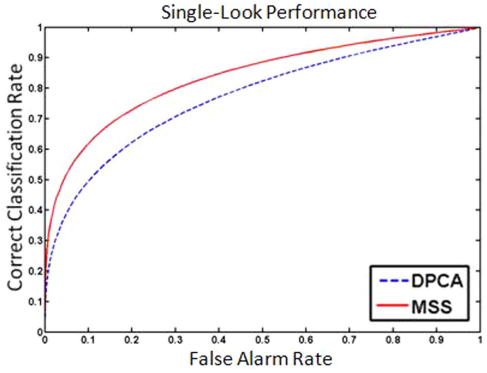


Fig. 23. Single-look ROC comparison.

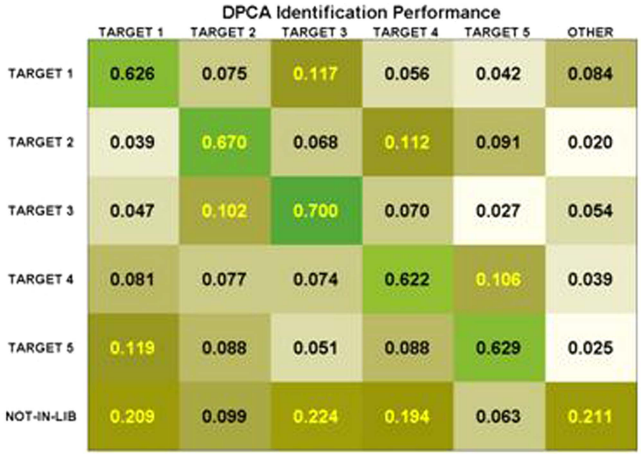


Fig. 21. DPCA single-look performance.

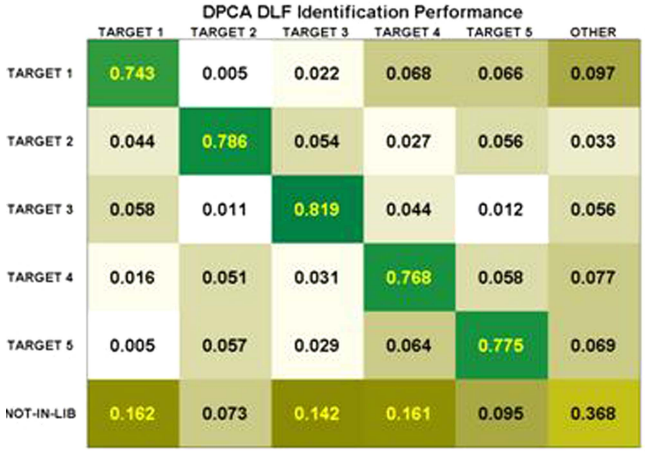


Fig. 24. DPCA decision-level fusion performance.

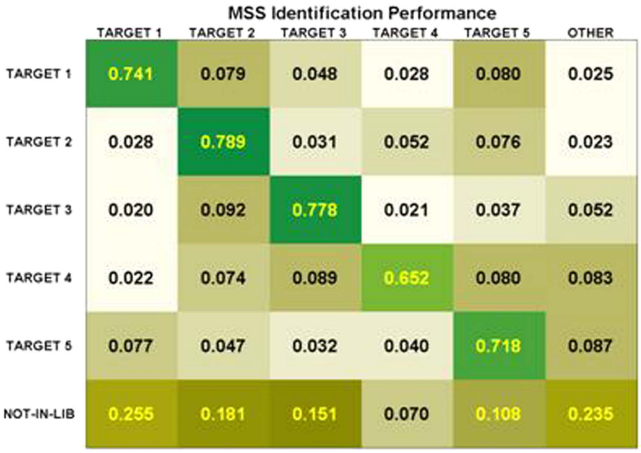


Fig. 22. Three-channel MSS single-look performance.

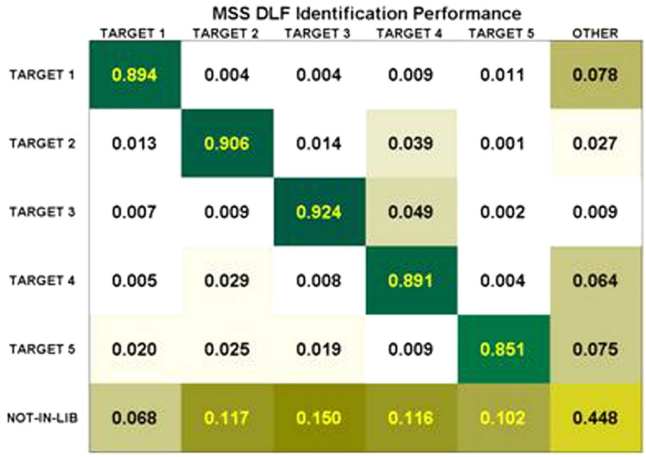


Fig. 25. Three-channel MSS decision-level fusion performance.

4.5. Performance Metrics

Fig. 27 shows the performance metrics computed using Equations 11–13 for each of the five in-library targets. Declaration, false alarm, and correct classification probabilities were compared for single-look DPCA (first blue bar), single-look MSS (second red bar), decision level fused multi-look DPCA (third yellow bar), and multi-look decision level fused MSS (fourth green bar) performance.

In general, target identification improves with enhanced clutter suppression and fused multi-look performance versus a single look ID. However, it is noted that not every example of MSS clutter cancellation results in improved individual target identification. An example of a false alarm increase was seen using the MSS in target 5. Using the DLF, both declaration and correct classification were improved. More importantly, with CM DLF, the false alarms were significantly reduced.

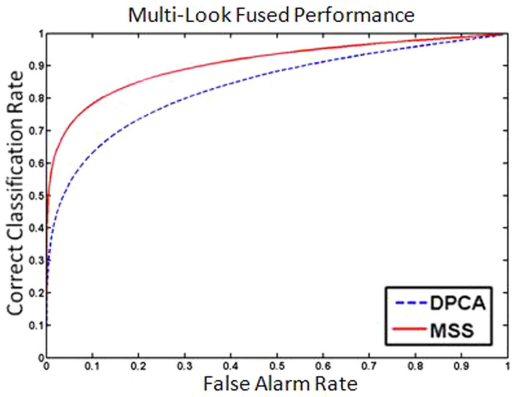


Fig. 26. Multi-look fusion ROC comparison.

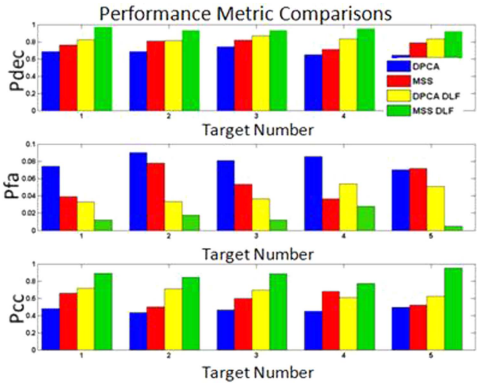


Fig. 27. DPCA and MSS single-look to multi-look performance metrics.

5. DISCUSSION AND CONCLUSIONS

The capability to collect and process three channels of radar data from a system configured with three phased-array antennas oriented in the along-track dimension has been demonstrated. The application of traditionally accepted two-channel clutter suppression techniques has been extended to true multi-channel data. The Multi-channel Signal Subspace (MSS) technique for two channels of data was demonstrated to be a superior clutter suppression technique to that of the Displaced Phase Center Antenna (DPCA) method. The MSS methodology was extended to exploit the additional information provided by the third channel of the phased array interrogating the scene.

The MSS technique applied to three channels of data suppressed the clutter well while preserving the features of the moving target. The signal-to-noise level of the three-channel MSS technique is approximately that of the two-channel MSS results. The availability of more target features in the range-Doppler image, while maintaining a good clutter suppression level, makes the MSS approach beneficial to automatic target recognition (ATR) applications. A significant ATR performance improvement is achieved with clutter suppressed data using the MSS algorithm relative to ATR performance with DPCA suppressed data.

A major factor not addressed in this paper; however, is that the processing time for the MSS algorithm is quite significant, especially when compared to the DPCA method. The processing times will need to be drastically reduced for the MSS algorithm to be practical in a data processing or operational environment. A potential solution is the parallelization of the time consuming block processing steps. This remains an area of future study.

The Confusion Matrix (CM) Decision-Level Fusion (DLF) multi-look technique improved target identification performance in comparison to the single-look ATR results. The CM-DLF algorithm performed extremely well with the MSS clutter suppressed data showing significant performance gains over both the single-look results and the DPCA fused performance. The DLF identification performance benefited from the target features preserved by the three-channel MSS clutter cancellation technique. For ATR applications, improved clutter suppression of HRR radar data greatly increases vehicle recognition and further enhances the resulting decision level fusion target identification of moving targets.

ACKNOWLEDGMENTS

The authors would like to thank Tim Ross for significant discussions and coordination in the Confusion Matrix DLF development.

REFERENCES

[1] D. Angelova and L. Mihaylova
Joint tracking and classification with particle filtering and mixture Kalman filtering using kinematic radar information.
Digital Signal Processing, **16**, 2 (Mar. 2006), 180–204.

[2] B. N. S. Babu, J. A. Torres, and W. L. Melvin
Processing and evaluation of multichannel airborne radar measurements (MCARM) measured data.
In *IEEE International Symposium on Phased Array Systems and Technology*, Oct. 1996.

[3] Y. Bar-Shalom and X-R. Li
Multitarget-Multisensor Tracking: Principles and Techniques.
YBS Publishing, Storrs, CT, 1995.

[4] V. Bhatnagar, A. K. Shaw, and R. W. Williams
Improved automatic target recognition using singular value decomposition.
In *Proceedings 1998 IEEE Conference on Acoustics, Speech, and Signal Processing*, vol. 5, 1998.

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

[6] E. Blasch
Derivation of A Belief Filter for High Range Resolution Radar Simultaneous Target Tracking and Identification.
Ph.D. Dissertation, Wright State University, 1999.

[7] E. Blasch
Assembling a distributed fused Information-based human-computer cognitive decision making tool.
IEEE Aerospace and Electronic Systems Magazine, **15**, 5 (May 2000), 11–17.

[8] E. Blasch and M. Bryant
Information assessment of SAR data for ATR.
In *Proceedings of IEEE National Aerospace and Electronics Conference*, 1998.

[9] E. Blasch and L. Hong
Data association through fusion of target track and identification sets.
In *International Conference on Information Fusion*, 2000.

[10] E. Blasch and S. Huang
Multilevel Feature-based fuzzy fusion for target recognition.
In *Proceedings of SPIE*, vol. 4051, 2000.

[11] E. Blasch, J. J. Westerkamp, J. R. Layne, L. Hong, F. D. Garber, and A. Shaw
Identifying moving HRR signatures with an ATR belief filter.
In *Proceedings of SPIE*, vol. 4053, 2000.

[12] E. Blasch and C. Yang
Ten methods to Fuse GMTI and HRRR measurements for joint tracking and identification.
In *International Conference on Information Fusion*, July 2004.

[13] L. E. Brennan and L. S. Reed
Theory of adaptive radar.
IEEE Transactions on Aerospace and Electronic Systems, **AES-9**, 2 (Mar. 1973), 237–252.

[14] L. A. Chan, N. M. Nasrabadi, and D. Torrieri
Discriminative eigen targets for automatic target recognition.
In *Proceedings of SPIE*, vol. 3371, 1998.

[15] L. A. Chan, N. M. Nasrabadi, and D. Torrieri
Eigenspace Transformation for automatic clutter rejection.
Optical Engineering, **40**, 4 (2001).

[16] H-C. Chiang, R. L. Moses, and L. C. Potter
Model based classification of radar images.
IEEE Transactions on Information Theory, **46**, 5 (Aug. 2000), 1842–1854.

[17] D. J. Choe and R. G. White
Moving target detection in SAR imagery: Experimental results.
In *IEEE International Radar Conference*, 1995, 644–649.

[18] F. Dicander and R. Jonsson
Comparison of Some HRR-classification algorithms.
In *Proceedings of SPIE*, vol. 4382, 2001.

[19] D. Gross, M. Oppenheimer, B. Kahler, B. Keaffaber, and R. Williams
Preliminary comparison of HRR signatures of moving and stationary ground vehicles.
In *Proceedings of SPIE*, vol. 4727, 2002.

[20] X. Guo and M. Soumekh
Signal subspace registration of time series medical imagery.
In *Proceedings of International Conference on Signal Processing*, 2002, 1524–1527.

[21] J. K. Jao
Theory of synthetic aperture radar imaging of a moving target.
IEEE Transactions on Geoscience and Remote Sensing, **39**, 9 (Sept. 2001), 1984–1992.

[22] B. Kahler and E. Blasch
Robust multi-look HRR ATR investigation through decision-level fusion evaluation.
In *International Conference on Information Fusion*, 2008.

[23] B. Kahler and E. Blasch
Impact of HRR radar processing on moving target identification performance.
In *International Conference on Information Fusion*, 2009.

[24] B. Kahler and E. Blasch
Target identification performance improvement from enhanced HRR radar clutter suppression.
In *Proceedings of the IEEE National Aerospace and Electronics Conference*, 2009.

[25] B. Kahler, E. Blasch, and L. Goodwon
Operating condition modeling for ATR fusion assessment.
In *Proceedings of SPIE*, vol. 6571, 2007.

[26] B. Kahler and B. Keaffaber
An improved multi-channel clutter suppression algorithm.
In *Proceedings of SPIE*, vol. 6568, 2007.

[27] B. Kahler, J. Querns, and G. Arnold
An ATR challenge problem using HRR data.
In *Proceedings of SPIE*, vol. 6970, 2008.

[28] R. Klemm
Introduction to space-time adaptive processing.
IEE ECEJ, special issue on STAP, **11**, 1 (Feb. 1999), 5–12.

[29] R. Klemm
Prospectives in STAP research.
In *Proceedings of the 2000 IEEE Sensor Array and Multichannel Signal Processing Workshop*, Mar. 2000, 7–11.

[30] G. Krieger, N. Gebert, and A. Moreira
Unambiguous SAR signal reconstruction from non-uniform displaced phase center sampling.
IEEE Geoscience and Remote Sensing Letters, **1**, 4 (Oct. 2004), 260–264.

[31] B. V. Kumar and M. Alkanhal
Eigen-extended maximum average correlation height (EEMACH) filters for automatic target recognition.
In *Proceedings of SPIE*, vol. 4379, 2001.

[32] J. Lancaster and S. Blackman
Joint IMM/MHT tracking and identification for multisensor ground tracking.
In *International Conference on Information Fusion*, 2006.

[33] B. Liu
Clutter suppression using recursive and non-recursive MTI filters.
IEEE Transactions on Aerospace and Electronic Systems, **24**, 3 (May 1988), 210–217.

[34] R. A. Mitchell and J. J. Westerkamp
Robust statistical feature based aircraft identification.
IEEE Transactions on Aerospace and Electronic Systems, **35**, 3 (July 1999), 1077–1094.

[35] C. E. Muehe and M. Labitt
Displaced-phase-center antenna technique.
Lincoln Laboratory Journal, **12**, 2 (2000).

[36] S. Mori, C-Y. Chong, E. Tse, and E. P. Wishner
Tracking and classifying multiple targets without a priori identification.
IEEE Transactions on Automatic Control, **AC-31**, 5 (May 1986), 401–409.

[37] W. E. Ng, C. B. Zhang, Y. H. Lu, and T. S. Yeo
Simulated DPCA performance for dual-channel SAR processing.
In *IEEE International Geoscience and Remote Sensing Symposium*, 1999.

[38] S. G. Nikolov, E. Fernandez Canga, J. J. Lewis, A. Loza, D. R. Bull, and C. N. Canagarajah
Adaptive image fusion using wavelets: Algorithms and system design.
In E. Lefebvre and P. Valin (Eds.), *Multisensor Data and Information Processing for Rapid and Robust Situation and Threat Assessment*, IOS Press, 2006.

[39] T. Nohara
Derivation of a 3-channel DPCA/monopulse radar using phased arrays.
IEEE National Telesystems Conference, 1994, 243–246.

[40] T. Nohara
Comparison of DPCA and STAP for space-based radar.
In *IEEE International Radar Conference*, 1995, 113–119.

[41] T. Nohara, P. Scarlett, and B. Eatock
A radar signal processor for space-based radar.
In *IEEE National Radar Conference*, 1993.

[42] L. M. Novak and G. J. Owirka
Radar target identification using an eigen-image approach.
In *IEEE Radar Conference*, 1994.

[43] L. M. Novak, G. J. Owirka, and A. L. Weaver
Automatic target recognition using enhanced resolution SAR data.
IEEE Transactions on Aerospace and Electronic Systems, **35**, 1 (Jan. 1999), 157–175.

[44] K. Ohba and K. Ikeuchi
Recognition of multi-specularity objects using the eigen-window.
In *IEEE 1996 Proceedings of the 13th International Conference on Pattern Recognition*, 1996, 692–696.

[45] S. Paul and A. K. Shaw
Robust HRR radar target identification by hybridization of HMM and eigen-template-based matched filtering.
In *Proceedings of SPIE*, vol. 5094, 2003.

[46] S. Paul, A. K. Shaw, K. Das, and A. K. Mitra
Improved HRR-ATR using hybridization of HMM and eigen-template-matched filtering.
In *IEEE Conference on Acoustics, Speech, and Signal Processing*, 2003.

[47] K. Ranney and M. Soumekh
Signal subspace change detection in averaged multi-look SAR imagery.
In *Proceedings of SPIE*, vol. 5808, 2005.

[48] P. G. Richardson
Analysis of the adaptive space time processing technique for airborne radar.
IEE Proceedings—Radar, Sonar Navigation, **141**, 4 (Aug. 1994), 187–195.

[49] P. G. Richardson
Effects of manoeuvre on space time adaptive processing performance.
In *IEEE Radar 97 Conference*, Pub. No. 449, Oct. 1997, 285–289.

[50] T. D. Ross, W. E. Pierson, E. G. Zelnio, and K. L. Priddy
Detection system performance metrics with scene content dependencies.
In *Proceedings of SPIE*, vol. 5808, 2005.

[51] K. Shaw and V. Bhatnagar
Automatic target recognition using eigen-templates.
In *Proceedings of SPIE*, vol. 3370, 1998.

[52] K. Shaw, R. Vashist, and R. Williams
HRR-ATR using eigen-template with noisy observations in unknown target scenario.
In *Proceedings of SPIE*, vol. 4053, 2000.

[53] W. Snyder, G. Ettinger, and S. Laprise
Modeling performance and image collection utility for multiple look ATR.
In *Proceedings of SPIE*, vol. 5427, 2004.

[54] M. Soumekh
Signal subspace fusion of uncalibrated sensors with application in SAR, diagnostic medicine and video processing.
IEEE Transactions on Image Processing, **8**, 1 (Jan. 1999), 127–137.

[55] M. Soumekh
Synthetic Aperture Radar with MATLAB Algorithms.
New York, NY: John Wiley & Sons, Inc., 1999.

[56] M. Soumekh
Moving target detection and imaging using an X band along-track monopulse SAR.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 1 (Jan. 2002), 315–333.

[57] M. Soumekh and B. Himed
SAR-MTI processing of multi-channel airborne radar measurement (MCARM) data.
In *Proceedings of IEEE Radar Conference*, 2002.

[58] M. Soumekh, S. Worrell, E. Zelnio, and B. Keaffaber
SAR wavefront reconstruction using motion compensated phase history (polar format) data and DPCA-based GMTI.
In *Proceedings of SPIE*, vol. 4053, 2000.

[59] W. Streilein, A. Waxman, W. Ross, F. Liu, M. Braun, D. Fay, P. Harmon, and C. H. Read
Fused multi-sensor image mining for feature foundation data.
In *International Conference on Information Fusion*, 2000.

[60] A. Tchamova, J. Dezert, T. Semerdjiev, and P. Konstantinova
Target tracking with generalized data association based on the general DSm rule of combination.
In *International Conference on Information Fusion*, 2004.

[61] J. D. Thompson
Verification of a decision level fusion algorithm using a proven ATR system and measured SAR data.
M.S. Thesis, Air Force Institute of Technology, 2006.

[62] R. Williams, J. Westerkamp, D. Gross, and A. Palomino
Automatic target recognition of time critical moving targets using 1D high range resolution (HRR) radar.
IEEE Aerospace and Electronic Systems Magazine, **15**, 4 (Apr. 2000), 37–43.

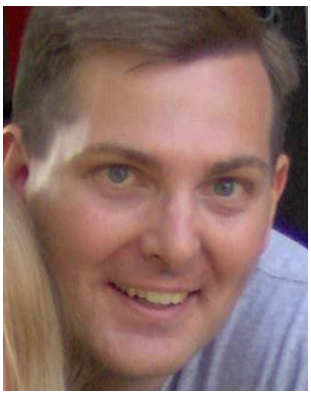
[63] R. Wu, Q. Gao, J. Liu, and H. Gu
ATR scheme based On 1-D HRR profiles.
Electronic Letters, **38**, 24 (Nov. 2002), 1586–1588.

[64] C. Yang and E Blasch
Mutual aided target tracking and identification.
Proceedings of SPIE, vol. 5099, 2003.

[65] C. Yang and E. Blasch
Pose angular-aiding for maneuvering target tracking.
International Conference on Information Fusion, 2005.

[66] C. Yang and E. Blasch
Pose-angular tracking of maneuvering targets with high range resolution radar (HRR).
International Conference on Information Fusion, 2008.

[67] C. Yang, E. Blasch, W. Garber, and R. Mitchell
A net track solution to pose-angular tracking of maneuvering targets in clutter with HRR radar.
IEEE Conference on Signals, Systems and Computers, 2007.



Bart Kahler received B.S. and M.S. degrees in electrical engineering from New Mexico State University in 1990 and 1992.

He researched high power microwave effects on the atmosphere and time domain scattering of electromagnetic energy while working for New Mexico State University and the Physical Science Laboratory. In 1993 he joined Intelligent Monitoring Systems and managed a diverse technical team developing real time environmental hazard monitoring equipment. The sensors developed used fluorescence detection to identify cholera bacteria and electro-chemical techniques to monitor heavy metals. He joined Mission Research Corporation in 1995 as a research engineer and contributed to the design and measurement of antennas and radomes for a variety of programs. He created modeling, analysis, and diagnostic tools to assist in hardware design and development, receiving the Mission Research Corporation EM Measurements Division Technical Achievement of the Year award in 1996 and 1997. As part of the Signature Exploitation Research Group, he used signal processing techniques with video phase history data for object identification, tracking, clutter suppression, and profile formation applications. In 2001 he joined General Dynamics as a Senior Systems Engineer, supporting signal processing, data exploitation, and information fusion efforts using infrared, electro-optical, and radar sensor data. Mr. Kahler developed and used sensor performance prediction models for a variety of systems and applied decision level fusion and feature level fusion techniques to improve object identification and tracking. In April 2010 he joined Science Applications International Corporation as a Sr. Radar Systems Engineer where he continues to explore signal processing and information fusion solutions for a number of customers. He is the recipient of a 2010 SAIC Jill Zimmerman Above and Beyond Award.

Mr. Kahler has published over 20 papers and 15 technical reports, and has received 10 corporate technical achievement awards. He received a performance award for the 1989 IEEE Digital Design Contest and was an organizer of the 1990 IEEE Student Professional Awareness Conference. He is a Senior Member of IEEE.

Erik Blasch received his B.S. in mechanical engineering from the Massachusetts Institute of Technology in 1992 and Master's degrees in mechanical engineering ('94), health science ('95), and industrial engineering (human factors) ('95) from Georgia Tech and attended the University of Wisconsin for a M.D./Ph.D. in mechanical engineering/neurosciences until being called to active duty in 1996 to the United States Air Force. He completed an M.B.A. ('98), M.S.E.E. ('98), M.S. econ ('99), M.S./Ph.D. psychology (ABD), and a Ph.D. in electrical engineering from Wright State University and is a graduate of Air War College.

He is currently a United States Air Force Research Laboratory (AFRL) exchange scientist to Defence R&D Canada (DRDC) at Valcartier, Quebec in the Future Command and Control (C2) Concepts and Structures Group of the C2 Decision Support Systems Section. Prior to the sabbatical, he was the information fusion evaluation tech lead for the AFRL Sensors Directorate–COMprehensive Performance Assessment of Sensor Exploitation (COMPASE) Center and an adjunct electrical engineering and biomedical engineering professor at Wright State University (WSU) and the Air Force Institute of Technology (AFIT) in Dayton, OH. He is also a reserve officer with the Air Force Office of Scientific Research (AFRL/AFOSR) in Arlington, VA.

Dr. Blasch was a founding member of the International Society of Information Fusion (ISIF) in 1998, a Member of the ISIF BoG from 2003 to 2010, and the 2007 ISIF president. He is currently a Member of the IEEE AESS BoG, associate editor for IEEE SMC-A, and Member of the IEEE AESS Track Standards Committee. He received the 2009 IEEE Russ Bioengineering Award and supported the IEEE 2005, 2008, and 2011 Sections Congress meetings. He began his career in robotics compiling over 30 top ten finishes as part of robotic teams in international competitions including winning the '91 American Tour del Sol solar car competition, '94 AAAI mobile robotics contest, and the '93 Aerial Unmanned Vehicle competition where they were first in the world to automatically control a helicopter. He has focused on automatic target recognition, targeting tracking, and information fusion research compiling 350+ scientific papers and book chapters. He has given numerous invited, keynote, and plenary talks as well as tutorials such as *Information Fusion Evaluation: from Methods to Metrics*. He is a Fellow of SPIE and a Senior Member of IEEE.

Estimating Network Parameters for Selecting Community Detection Algorithms

LETO PEEL

This paper considers the problem of algorithm selection for community detection. The aim of community detection is to identify sets of nodes in a network which are more interconnected relative to their connectivity to the rest of the network. A large number of algorithms have been developed to tackle this problem, but as with any machine learning task there is no "one-size-fits-all" and each algorithm excels in a specific part of the problem space. This paper examines the performance of algorithms developed for weighted networks against those using unweighted networks for different parts of the problem space (parameterised by the intra/inter community links). It is then demonstrated how the choice of algorithm (weighted/unweighted) can be made based only on the observed network.

Manuscript received October 26, 2010; revised May 05, 2011; released for publication June 11, 2011.

Refereeing of this contribution was handled by Mujdat Cetin.

Author's address: Advanced Technology Centre, BAE Systems, Bristol, UK, E-mail (leto.peel@baesystems.com).

1557-6418/11/$17.00

1. INTRODUCTION

The study of large scale networks has revealed a number of properties about the behaviour and topology of naturally occurring networks. One such property is the presence of community structures; sets of nodes in a network which are more interconnected relative to their connections to the rest of the network. The aim of community detection is to identify these structures. Community detection is a problem which has attracted much interest in recent years [7], [12], [19], [23], [27] and has consequently produced a wide range of approaches to the problem; an in-depth review of most contemporary methods is given in [8].

One of the reasons why the ability to detect communities is so attractive lies in the phenomenon known as *assortative mixing*, where entities in a network are observed to associate preferentially with similar entities. This suggests that detecting communities may be used for identifying entities which share common attributes or purposes. An example of community structures corresponding to entity similarity is given in [20] where community structures in a friendship network correspond to similarities in race and age. The wide range of complex systems that can naturally be expressed as networks (human interaction patterns, metabolic networks, WWW, and the brain) implies that community detection has applications spanning domains as diverse as biology [11], [14], [29], sociology [2], [11], [28], computer science [26], [30] and intelligence [1], [10], [16].

The implications of community detection in the intelligence domain are that it could be used to identify groups of people who share common goals or purposes. To this effect, community detection could potentially be used to constrain the inference problem when investigating or detecting malicious activities, e.g. rather than monitoring all people, use community detection as a pre-processing step to select a subset of people to monitor. In this setting, the network nodes would represent people and the links would represent interactions or relationships between them; such a network can be constructed from a database of phone records, email logs or other transactional data.

With a large selection of algorithms available to undertake the task of community detection, choosing an appropriate algorithm becomes problematic. This is largely due to the lack of formal or commonly accepted evaluation procedures. The networks used to evaluate community detection tend to be a small selection of real networks and/or networks generated from simple models, where these networks vary widely between authors. Recent work to address this has focused on developing benchmark networks [17] on which comparative analysis [18] can be drawn to determine the reliability of different algorithms. However, it is commonly accepted across the machine learning community that there is no *one-size-fits-all* solution and so this work considers the idea that for different situations, different classes

of algorithms may outperform other classes of algorithms. The range of community detection algorithms in itself poses the intelligence analyst with the challenge of choosing an appropriate solution or combinations of solution techniques for the specific problem at hand. It is therefore desirable to be able to provide the intelligence analyst, who will likely not have expert knowledge of these algorithms, with appropriate guidance. This paper considers the problem of automatically selecting community detection algorithms based on observations of the community structure.

The algorithm selection problem can be solved in part by design. For example, consider the class of algorithms involving modularity [22] optimisation. Simulated annealing [15] produces good results, but for large networks it may be preferable to substitute performance for speed by using a greedy algorithm such as in [5]. In this situation the trade off is straight forward, a choice of speed vs. quality. However, some algorithms address other limitations for which the trade off is not so clear cut. For example the modularity function suffers from a resolution limit [9] meaning that in some networks several communities become merged into one (e.g. communities consisting of two cliques connected by a single link). Some algorithms [3], [4] address this issue but it is not clear what weaknesses (if any) this might introduce. In cases such as these further investigation is required to determine the appropriate algorithm. In this work the class of algorithms which incorporate link weight information is examined to determine when these are most appropriate.

It has been previously observed how structural properties of communities affect the performance of community detection algorithms [18]. These properties cannot be measured from the network data alone as they require knowledge of the underlying community assignment. The main contribution of this work is to demonstrate how these structural properties can be estimated from features of the observed network. Therefore a prediction about which algorithm will perform best can be made. This is achieved by considering algorithms for weighted networks and algorithms for unweighted networks as two separate classes and demonstrating how the performance of these two classes differs across the problem space (defined in Section 2). Finally, a Support Vector Machine (SVM) [6] is used to classify the networks according to the algorithm which will perform best.

The rest of the paper is organised as follows: Section 2 defines the problem space by defining the network and community structure types and the target algorithm classes. The performance of the algorithm classes with respect to the structural parameters is evaluated in Section 3. Section 4 describes the *observable* network parameters and how a mapping can be made from these to the underlying structural parameters. The results of using the observable parameters to choose an appropriate class of algorithms are given in Section 5. Conclusions are given in Section 6.

2. PROBLEM SPACE

A network is a structure made up of nodes, representing entities, and links or edges, representing relationships or interactions between entities. The total number of links connected to a node is known as its degree. The network links may also have weights associated with them which may represent the relative importance of the link. For example, in an interaction network representing a phone record database, the nodes would represent people and the links phone calls. The link weights could then represent the frequency of calls. Network links may also be directed, but this will not be considered in this work.

The premise of community detection is that there is some underlying assignment of nodes to communities which has to be discovered. But despite the large amount of literature on the subject there is still a lack of agreement on what defines a community beyond the intuitive concept that community structures have more intra-community links than inter-community links. Without a common definition it is difficult to draw a comparison between algorithms. However, it may not be necessary (or even desirable) to define a specific common definition of community, as definitions may be dependent on the application. Instead, perhaps all that is required is a suitably comprehensive parameter set for describing the space of community types and structures of interest.

A reasonable starting point is the parameter set used to generate networks and communities using the Lancichinetti-Fortunato-Radicchi (LFR) benchmark generator [17] as not only do these describe a number of network properties, but by using the generator it is possible to obtain networks and community assignments with those properties. This parameter set is described in Section 2.1.

2.1. Network-Community Parameterisation

The parameter set used to describe the problem space are the parameters used by the LFR benchmark which is fully described in [17]. The LFR benchmark was designed to generate datasets to test community detection algorithms and mimic the observed properties of large-scale real complex networks [21], such as power-law degree and community distribution.

The parameters are best described in the context of the graph generation procedure:

1. N nodes are assigned to communities such that the community size distribution conforms to a power-law with minus exponent τ_2.

2. Each node is assigned a degree such that the degree distribution conforms to a power law with minus exponent τ_1 and mean degree k.

3. Links are initially assigned randomly according to the degree distribution. A topological mixing parameter, μ_t, is set to define the proportion of each node's links which link outside its community. Topological consistency with this parameter is achieved through an iterative re-wiring procedure.

4. Each node is then assigned a strength according to a power-law distribution with minus exponent β. The strength of a node is the weighted analogy of degree and as such represents the sum of the weights of the links for a given node.

5. To assign the link weights a similar process to step 3 is carried out according to the weight mixing parameter, μ_w.

Networks and their communities can be generated for a specified set of parameter values using the above procedure. In addition other constraining parameters can be specified including the maximum degree ($\max k$), and the minimum and maximum sizes of communities ($c\min$ and $c\max$ respectively). Using these parameters to describe the problem space has the advantage that by using the generative procedure outlined above it is possible to obtain a network for a given set of parameter values. It is accepted that these may not be a full set of parameters to comprehensively describe the space of all possible network-community structures. Even so, the space is one of high dimensionality and so full exploration of all the parameters is beyond the scope of this paper and remains for future work. To constrain the problem, the values of all parameters were fixed with the exception of μ_t and μ_w, which from initial tests were found to have the greatest impact on use of link weights.

2.2. Algorithm Overview

The algorithm selection problem has been constrained to choosing between the class of algorithms which use link weight information and the class that does not. In light of this, it was decided to use algorithms suitable for unweighted or weighted networks. This way a controlled comparison can be drawn between the performances of the unweighted and weighted algorithms without being concerned about differences in algorithms. Two such algorithms are examined:

- *Infomap* [27]: This algorithm approaches the community detection problem by identifying a duality between community detection and information compression. By using random walks to analyse the information flow through a network it identifies communities as modules through which information flows quickly and easily. Coding theory is used to compress the data stream describing the random walks by assigning frequently visited nodes a shorter codeword. This is further optimised by assigning unique codewords to network modules and reusing short codewords for network nodes such that node names are unique given the context of the module. This two level description of the path allows a more efficient compression by capitalising on the fact that a random walker spends more time within a community than moving between communities.
- *COPRA* [13]: This is an extension of the label propagation based RAK algorithm [25]. The algorithm works as follows; to start, all nodes are initialised with a unique label. These labels are then updated iteratively, where a node's new label is assigned according to the label used most by its neighbours. If there is more than one most frequently occurring label amongst the neighbours, then the label is chosen randomly. At termination of the algorithm, nodes with the same label are assigned to the same community. The **C**ommunity **O**verlap **PR**opagation **A**lgorithm (COPRA) extends the RAK algorithm to deal with the possibility of overlapping communities (although this aspect of community detection is not explored within this work). This is done by augmenting the label with a belonging factor such that for a given node these sum to 1. To prevent all nodes becoming a member of all communities, a threshold is set below which the labels are discarded. Due to the stochastic nature of the algorithm, the algorithm is run a number of times and the "best" community assignment is decided according to the one which has the highest modularity [22]. In the weighted instance of the algorithm, the weights of the network are incorporated by weighting the frequency of the labels according to the link weight connecting the respective node.

3. ALGORITHM PERFORMANCE

A number of different metrics are used in the literature to measure the performance of community detection algorithms, however the Normalised Mutual Information [19] metric is one which has become fairly standard recently and so will be used here. This metric provides a measure of similarity between the algorithm output assignment and the true community assignment, where a value of 1 denotes a perfect match. Using the networks described in Section 2.1, experiments were run to examine the effect of varying the two mixing parameters μ_t and μ_w, the results of which can be seen in Fig. 1.

Fig. 1 shows the mutual information scores for the weighted algorithms (COPRAw, INFOMAPw) and unweighted algorithms (COPRAuw, INFOMAPuw) as μ_w is changed. The plots (a)–(d) show the performance for different values of μ_t. Each point on the graphs represents the average mutual information over 100 generated networks (using the LFR benchmark generator described in Section 2.1) with the indicated parameter values. It can be seen that the unweighted algorithms perform well when μ_t is low and are unaffected by μ_w for all values μ_t. This is because these algorithms only

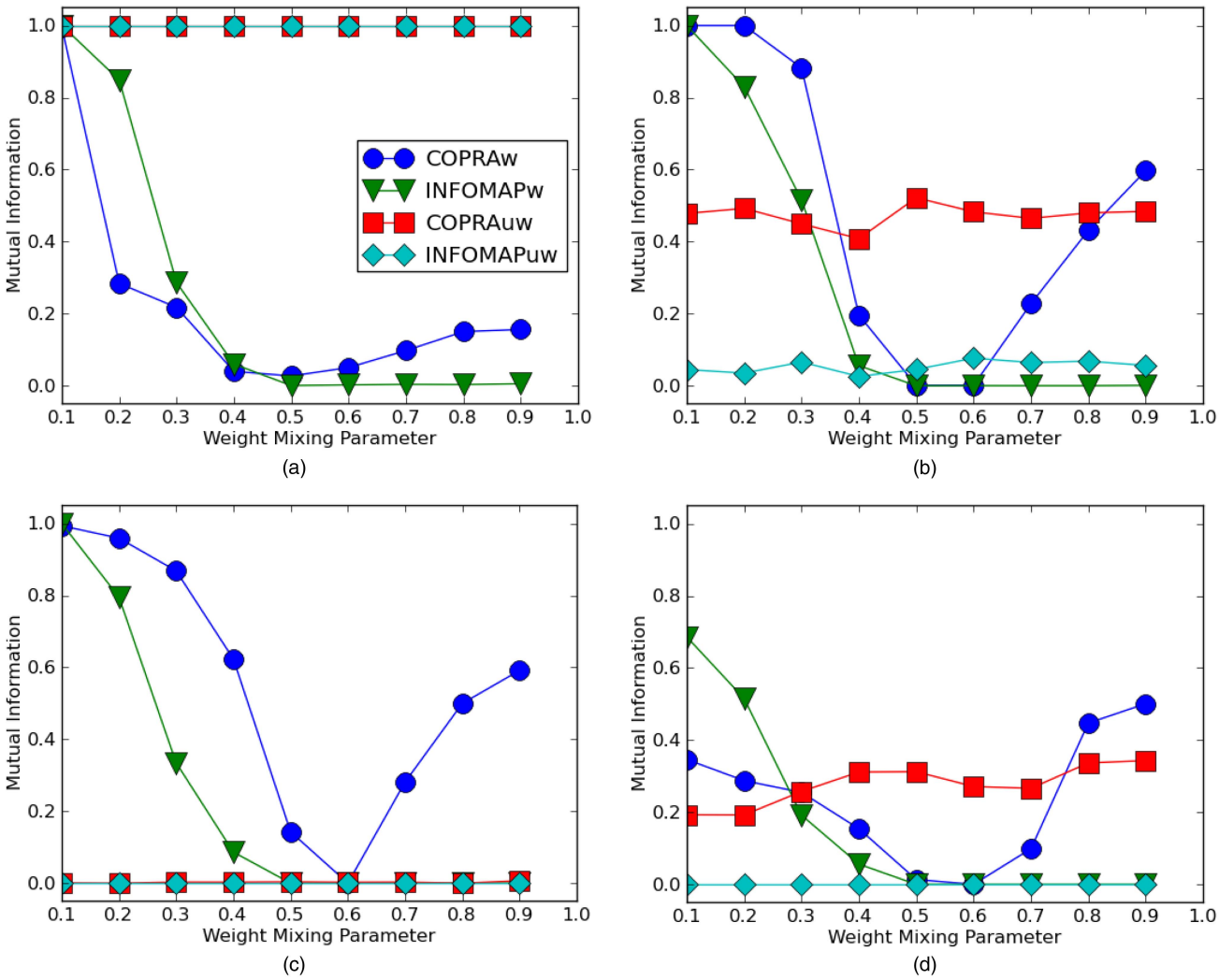


Fig. 1. Mutual information scores (y-axis) as μ_w (x-axis) changes. Each subplot shows a different fixed value for μ_t. The values of the other parameters were fixed: $N = 100$, $k = 25$, max$k = 50$ $\tau_1 = 2$, $\tau_2 = 1$, $\beta = 1.5$, $c$min $= 15$, $c$max $= 50$. (a) $\mu_t = 0.2$. (b) $\mu_t = 0.4$. (c) $\mu_t = 0.6$. (d) $\mu_t = 0.8$.

rely on the topological information. The weighted algorithms on the other hand are affected by both parameters, but are seen to consistently perform well for low μ_w. The effect of μ_t is probably best observed in Fig. 2. Here it can be seen that the weighted algorithms perform well when μ_t is at least as high as μ_w (in this case $\mu_w = 0.3$). A similar observation was made in [18] where it was seen that weighted algorithms performed better overall at μ_t values of 0.5, in comparison to lower values. It was explained that the reason for this is that a low μ_t relative to μ_w means that there is a lower proportion of inter-community links relative to the proportion of inter-community weights. The effect of this is that a small number of inter-community links receive high link weights relative to the intra-community weights, see Fig. 4.

The effect of this is that there are regions of the problem space, parameterised by community mixing proportions, in which a weighted algorithm will outperform an unweighted one and vice versa. This can be seen in Fig. 2 where the two regions are labelled w (weighted) and uw (unweighted). This result indicates that a choice can be made, based on the community structure, as to the class of community detection algorithms.

In order to take advantage of this information and select the best class of algorithms for a given network, some knowledge of the underlying community structure is required. It may be possible to make some assumption about the communities that are sought after based on some knowledge of the specific domain. In most community detection problems however, this information about the community structure is unknown.

4. PARAMETER ESTIMATION

In order to use the information from the previous section, it is required to know the values of the mixing parameters of the communities. Without knowledge of the communities (i.e. prior to community detection) it is not possible to evaluate these parameters. In this section it will be shown how parameters of the observable

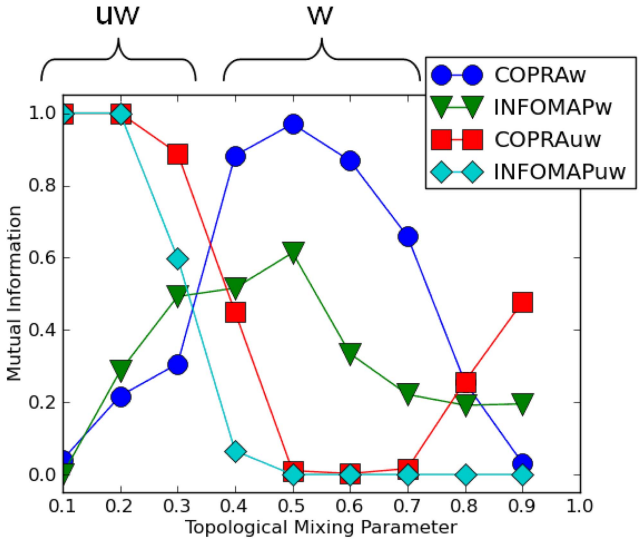


Fig. 2. Mutual information scores for the weighted (w) and unweighted (uw) algorithms as μ_t is varied. The value of μ_w is fixed at 0.3. It is noticeable that the two classes of algorithm perform for complimentary settings of μ_t.

network can be mapped to these community parameters and how these values can be used to build a classifier to determine the class of community detection most suitable for the given network.

4.1. Observable Parameters

There are a range of metrics associated with describing network topology: degree distribution, average diameter, and centrality measures are a few of them. The problem here is that a parameter is required which describes the way that the community structures interact, without explicitly knowing the community structures.

To approach this, the node measure called clustering coefficient [31] is considered. This is defined as:

$$C^{(v)}_{\text{unweighted}} = \frac{\sum_{i,j \in N_v} e_{ij}}{k_v(k_v - 1)/2} \qquad (1)$$

where the local clustering coefficient, $C^{(v)}_{\text{unweighted}}$, represents the proportion of the neighbours, N_v, of node v which are connected (i.e. edge $e_{ij} = 1$ if there is a link between neighbouring nodes i and j) out of the possible connections between its neighbours, $k_v(k_v - 1)/2$.[1] It was found that the mean value of the local clustering coefficient, taken over all the nodes in the network, showed a strong correlation with the topological mixing parameter, μ_t (Fig. 3(a)). This is similar to an observation in [24] where high clustering coefficients were observed in Girvan-Newman networks [11] with low ratios of inter- to intra-modular connectivity (a parameter

[1] If the degree of a node is less than or equal to 1 then the clustering coefficient is defined as 0.

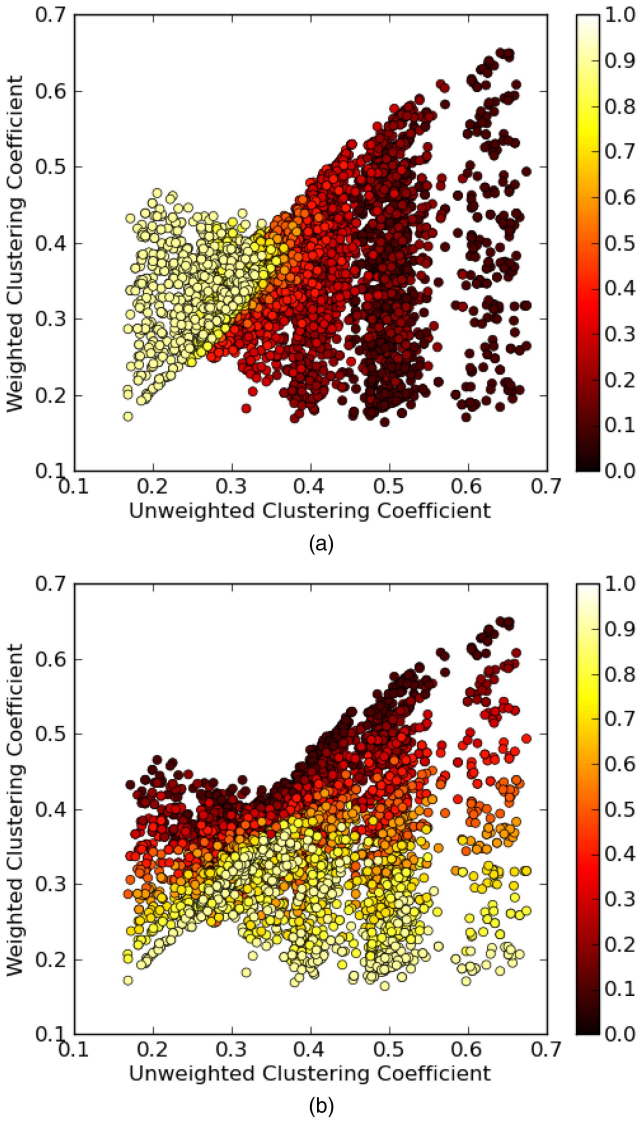


Fig. 3. Scatter plots of the unweighted (x-axis) and weighted (y-axis) mean local clustering coefficient for the networks in Fig. 1, (a) shows the value of the topology mixing parameter, μ_t. Similarly (b) shows the weight mixing parameter, μ_w.

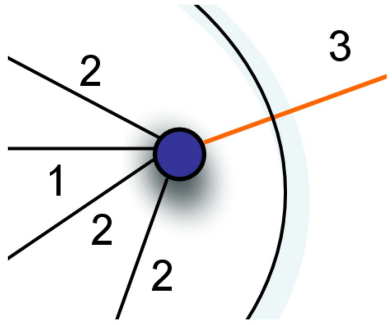


Fig. 4. An example node with links and weights from a network with $\mu_t = 0.2$ and $\mu_w = 0.3$. As a result the single inter-community link (orange) receives a higher weight relative to the intra-community links.

which bears resemblance to the topological mixing parameter, μ_t, in the more realistic LFR networks used here).

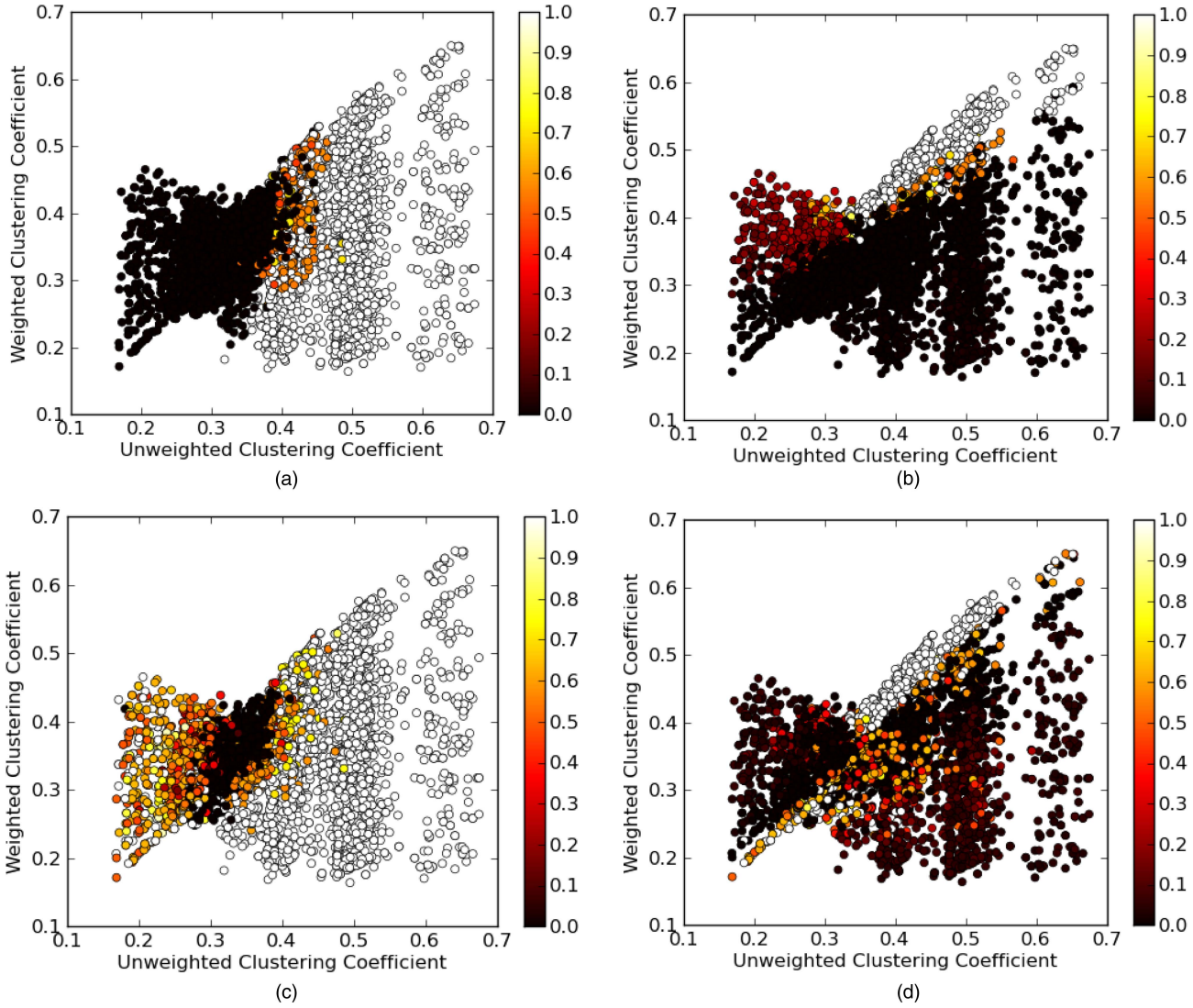


Fig. 5. Clustering coefficients scatter plots (the same networks used in Fig. 1) with colours showing the mutual information score for (a) unweighted infomap, (b) weighted infomap, (c) unweighted COPRA and (d) weighted COPRA.

This observation suggests that the mean clustering coefficient could be used to estimate the mixing parameter μ_t. If the mean clustering coefficient could be used to estimate the topological mixing then it follows that a weighted extension to this may yield information about the weighted mixing parameter (Equation 2).

$$C^{(v)}_{\text{weighted}} = \frac{\sum_{i,j \in N_v} (w_{vi} + w_{vj}) e_{ij}}{\sum_{i \in N_v} w_{vi} (k_v - 1)} \tag{2}$$

where w_{vi} is the weight associated with the link between nodes v and i. The mean of this value over the network was found to correlate with μ_w (Fig. 3(b)). The results in Fig. 3 suggest that the mixing parameters can be estimated from observed network characteristics without knowledge of the community structure.

The reason for this can be explained by considering the general principle of a community; that nodes within a community are more likely to be connected compared to overall probability of connection due to the sparse nature of the network. Hence, if two neighbours are within the same community, it is reasonable to expect them to be connected. However, if neighbours are not in the same community it is more likely that they are not connected. Based on this reasoning, the local clustering coefficient is an estimate of the individual node's mixing parameter, which averaged over the network yields a global estimate.

4.2. Algorithm Classification using SVM

The results of the previous section suggest that it is possible to estimate the mixing parameters of the communities. Now returning to the reason why it may be useful to estimate these parameters, i.e. to determine the class of algorithm, it is suggested that rather than estimate the mixing parameters and in turn predict the algorithm class, it may be more useful to use the clustering coefficients to directly predict the algorithm class. Fig. 5 shows similar plots as Fig. 4, but with the colour indicating the performance for the different

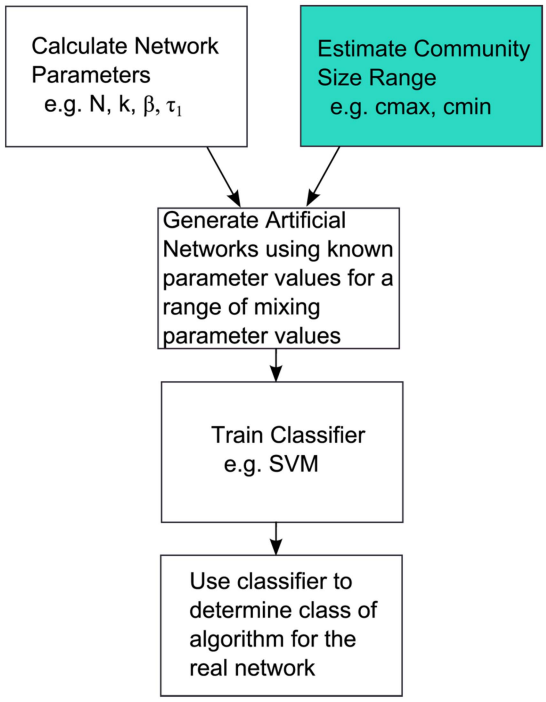


Fig. 6. Algorithm selection pipeline for classification of real networks. The shaded box indicates a step which requires further work to define.

algorithms. It can be seen that the weighted algorithms have a distinctly different performance pattern from the unweighted ones.

In order to confirm that these observable parameters can effectively predict the algorithm class, a simple classifier was built using support vector machines (SVM) [6] with a Gaussian kernel. To do this, each of the networks was assigned a class weighted, unweighted, none based on the class of algorithm which performed best in terms of its mutual information score. A class of "none" was assigned to any network where the mutual information score for the best performing algorithm was below some threshold. The reasoning for this is that for low performance values the output is not meaningful and therefore the choice of algorithm is irrelevant. As SVMs are restricted to two classes, three classifiers were trained (weighted vs. unweighted, weighted vs. none, unweighted vs. none) and the predicted class obtained by using a voting scheme over the three outputs, i.e. the predicted class is the most frequent output of the three classifiers. The results are discussed in Section 5.

4.3. Applicability to Other Networks

So far all the networks examined have had the same network properties except for the mixing parameters. To investigate if this relationship still holds for networks with different parameter settings, the experiments were repeated for: larger networks ($N = \{1000, 5000\}$), small and large community sizes ($c\min = \{15, 25\}$, $c\max = \{50, 100\}$), and different average degree ($k = \{25, 35\}$).

TABLE I
Classifier Confusion Matrix (100 nodes)

		Predicted Class		
		Weighted	Unweighted	None
True Class	Weighted	160	6	146
	Unweighted	8	369	18
	None	29	51	1213

Fig. 7 shows the clustering coefficient scatter plots indicating the mixing parameter values of the different graphs at each of the parameter settings. It can be seen that as long as the other parameter values are fixed there is still a correlation between the unweighted clustering coefficient and topological mixing parameter μ_t and between the weighted clustering coefficient and weight mixing parameter μ_w. It can also be seen from the scatter plots that these relationships are more or less the same for most of the different parameter settings. The main differences occur when the community size changes relative to the number of nodes in the network, i.e. as the ratio of community size to network size increases the points on the scatter plots become more compressed.

In order to apply the algorithm classification method presented in this paper to real networks, a classifier could be trained on a set of artificial networks which resemble the real network. This process is outlined in Fig. 6. There are a number of limitations with this method. First, the step which estimates the community size range (indicated by the shaded box) is currently undefined and requires further investigation to identify a method for estimating this range prior to detecting communities. Second, the process relies on the assumption that the artificial networks used for training are representative of the real network.

5. RESULTS

A SVM with a Gaussian kernel was trained on 8000 100 node networks (described in Section 3) taking the unweighted and weighted mean clustering coefficients as inputs. The "none" class was defined as networks for which the maximum mutual information score was below 0.6. The output classes for the test set (2000 networks) are displayed in Fig. 9. This can be compared to the true class labels in Fig. 8. The overall performance on the test set was 87.1%. A confusion matrix of the test set performance is shown in Table I.

To confirm these results, Fig. 10 shows the mean performance, according to mutual information, when selecting the algorithm class using this classifier. This is compared against the performance of the best weighted algorithm and the best unweighted algorithm.

Table II shows the confusion matrix for the same experiment run on all the networks described in Section 4.3 where the number of nodes in the network equals 1000 (11966 training networks, 17955 test net-

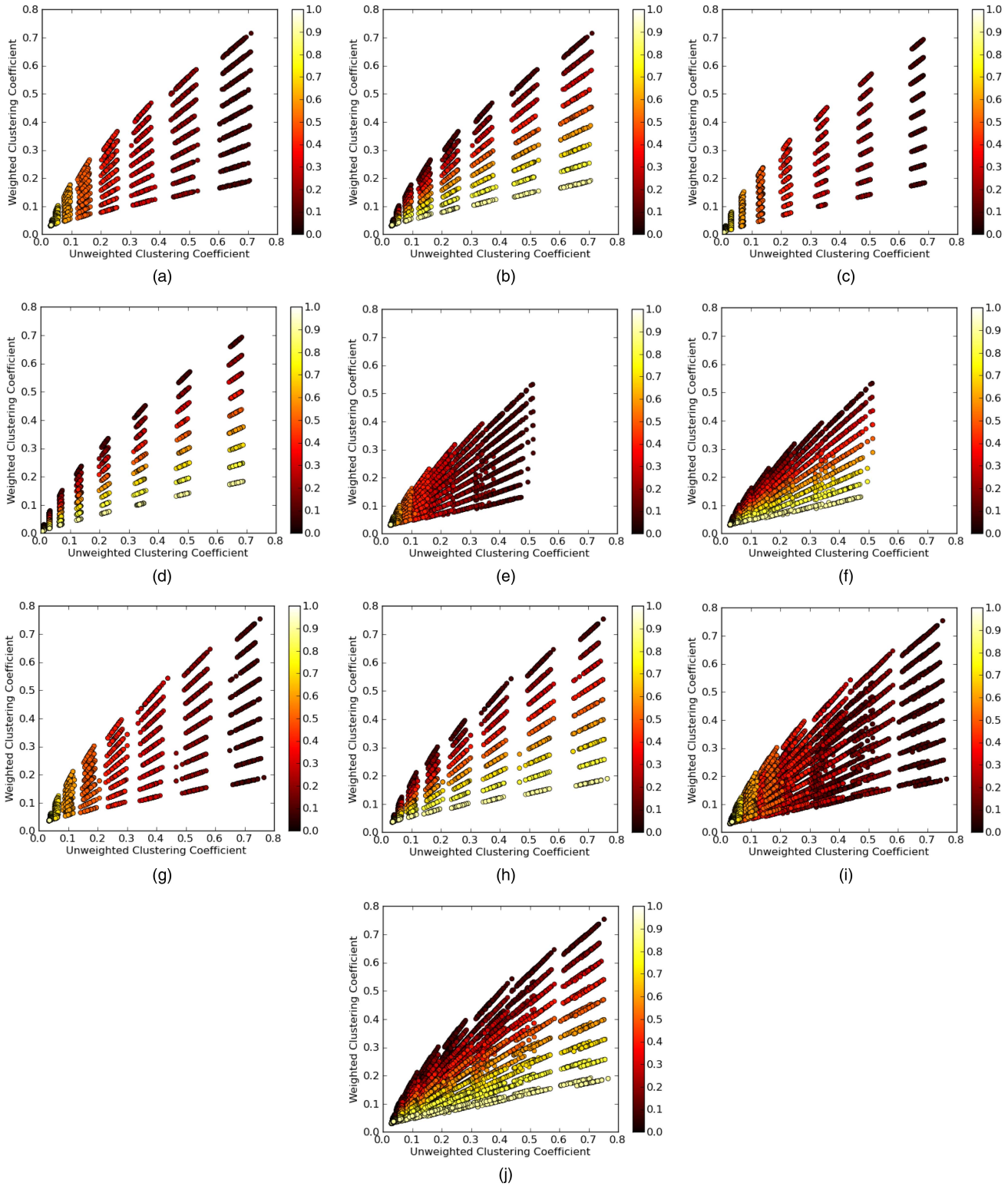


Fig. 7. Scatter plots of the unweighted (x-axis) and weighted (y-axis) mean local clustering coefficient for networks with different parameter settings, the left column shows the value of the topology mixing parameter, μ_t. Similarly the right column shows the weight mixing parameter, μ_w. (a) N = 1000, k = 25, $c$min = 15, $c$max = 50 (topology mixing parameter, μ_t). (b) N = 1000, k = 25, $c$min = 15, $c$max = 50 (weight mixing parameter, μ_w). (c) N = 5000, k = 25, $c$min = 15, $c$max = 50 (topology mixing parameter, μ_t). (d) N = 5000, k = 25, $c$min = 15, $c$max = 50 (weight mixing parameter, μ_w). (e) N = 1000, k = 25, $c$min = 25, $c$max = 100 (topology mixing parameter, μ_t). (f) N = 1000, k = 25, $c$min = 25, $c$max = 100 (weight mixing parameter, μ_w). (g) N = 1000, k = 35, $c$min = 15, $c$max = 50 (topology mixing parameter, μ_t). (h) N = 1000, k = 35, $c$min = 15, $c$max = 50 (weight mixing parameter, μ_w). (i) All networks where N = 1000 (topology mixing parameter, μ_t). (j) All networks where N = 1000 (weight mixing parameter, μ_w).

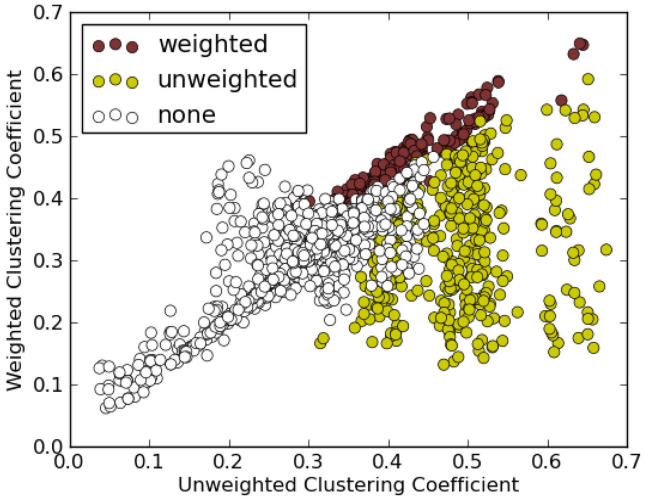


Fig. 8. The true classification of the 100 node networks in the test set.

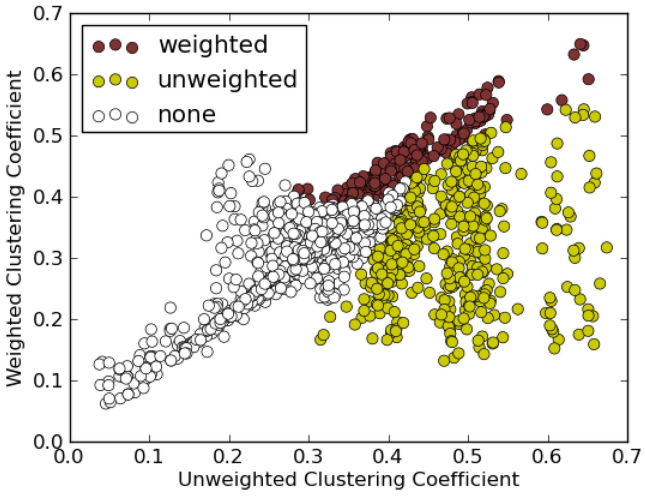


Fig. 9. The predicted classification of the 100 node networks in the test set using a SVM classifier.

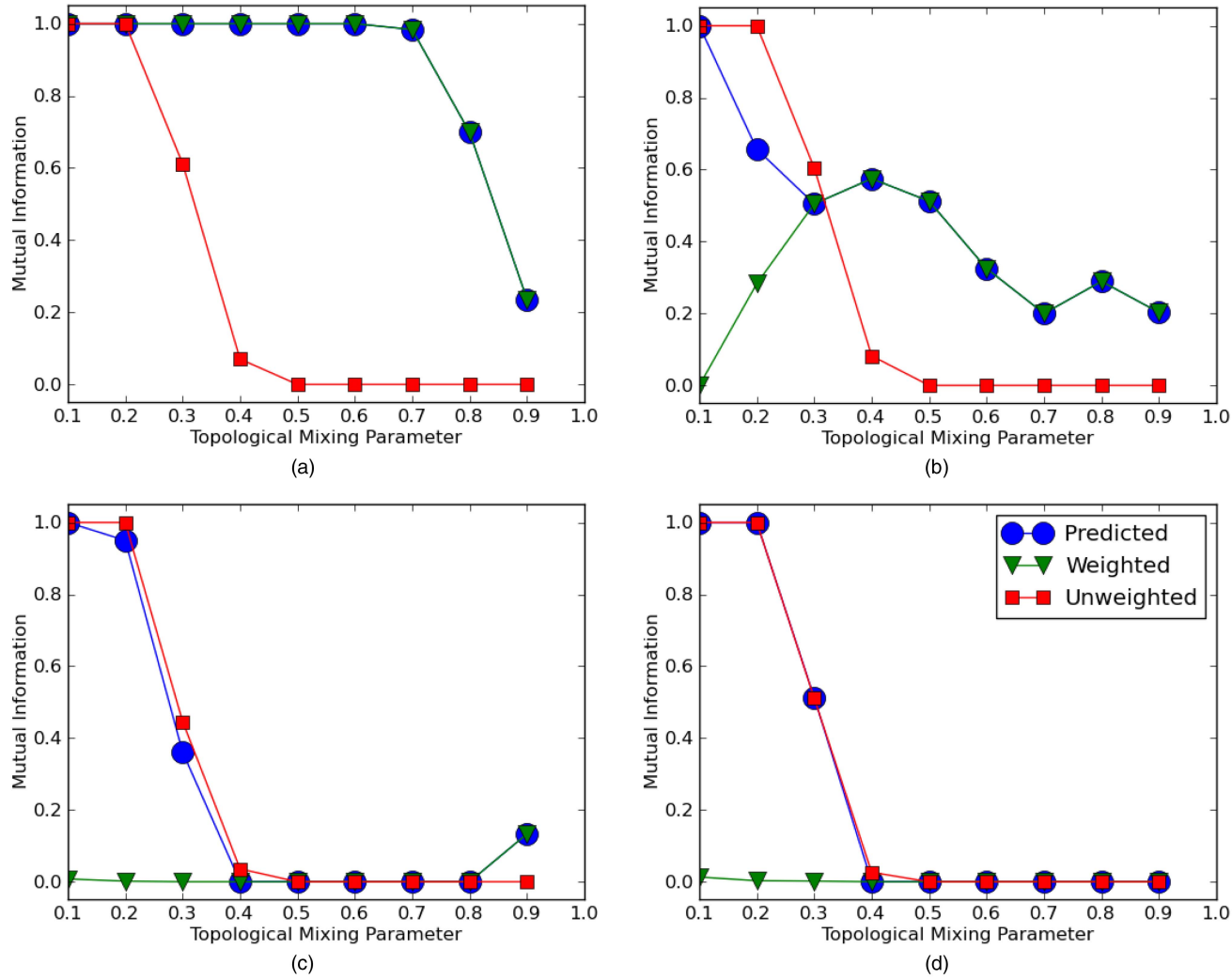


Fig. 10. Mutual information scores on 100 node networks for when algorithm class is selected by the classifier compared to the individual algorithm classes. Each graph shows the performance (y-axis) as μ_t (x-axis) is varied for different μ_w values. (a) $\mu_w = 0.1$. (b) $\mu_w = 0.3$. (c) $\mu_w = 0.5$. (d) $\mu_w = 0.7$.

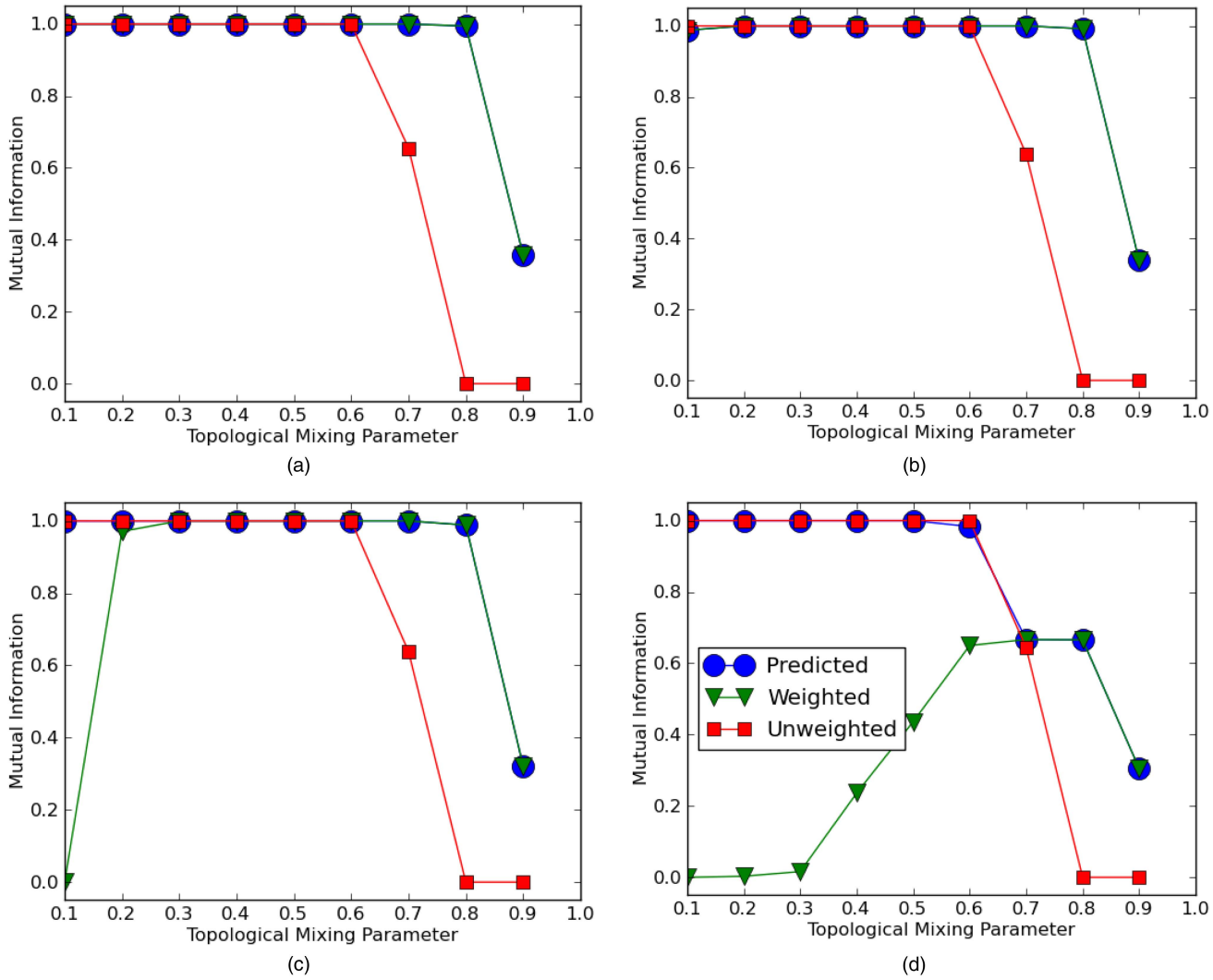


Fig. 11. Mutual information scores on all networks of 1000 nodes for when algorithm class is selected by the classifier compared to the individual algorithm classes. Each graph shows the performance (y-axis) as μ_t (x-axis) is varied for different μ_w values. (a) $\mu_w = 0.1$. (b) $\mu_w = 0.3$. (c) $\mu_w = 0.5$. (d) $\mu_w = 0.7$.

TABLE II
Classifier Confusion Matrix (1000 nodes)

		Predicted Class		
		Weighted	Unweighted	None
True Class	Weighted	7650	477	467
	Unweighted	85	5237	0
	None	320	17	3702

works). Similarly, Fig. 11 shows the mean performance when selecting the algorithm class using this classifier. From these graphs it can be seen that the classifier is able to select an appropriate class of algorithm such that it can achieve near optimum performance, constrained by the algorithms considered.

From these results it can be seen that even with a simple classifier it is possible to obtain accurate predictions for the best class of community detection algorithm based on properties of the network alone.

6. CONCLUSION

To the best of the author's knowledge, no previous work has explored the problem of choosing an appropriate community detection algorithm based on the underlying structural properties. This work has presented community detection algorithms as examples of two classes of algorithm: weighted or unweighted. It is demonstrated that for different types of network and community structure, the class of algorithm has an effect on the performance. It has been shown that for some weighted networks, unweighted algorithms can produce better results than their weighted counterparts. Furthermore it has been shown that it is possible to choose the algorithm class based only on the observed network parameters without prior knowledge of the community structure or assignment.

The algorithm selection demonstrated in this work relies on the training networks (and their communities) to share similar properties as the test network(s). In particular the community size in relation to the size of the

network is shown to have an effect on the relationship between the observable clustering coefficients and the underlying mixing parameters. It is left to future work to identify a means to determine this ratio of community to network sizes.

In addition, the algorithm selection in this work is constrained in terms of the classes of algorithms considered. Future work will reduce these constraints by considering a more comprehensive set of algorithms and classes.

ACKNOWLEDGMENTS

This work was undertaken as part of an Engineering Doctorate at the EngD VEIV Centre for Doctoral Training at University College London. UK Patent Pending No.: 1004376.8, March 2010.

REFERENCES

[1] J. Adibi, H. Chalupsky, E. Melz, and A. Valente
The KOJAK group finder: Connecting the dots via integrated knowledge-based and statistical reasoning.
In *Proceedings of the national conference on Artificial Intelligence*, 2004, 800–807.

[2] A. Arenas, L. Danon, A. Diaz-Guilera, P. M. Gleiser, and R. Guimera
Community analysis in social networks.
The European Physical Journal B-Condensed Matter and Complex Systems, **38**, 2 (2004), 373–380.

[3] A. Arenas, A. Fernández, and S. Gómez
Analysis of the structure of complex networks at different resolution levels.
New Journal of Physics, **10** (2008), 053039.

[4] V. D. Blondel, J-L. Guillaume, R. Lambiotte, and E. Lefebvre
Fast unfolding of communities in large networks.
Journal of Statistical Mechanics: Theory and Experiment, (Oct. 2008), P10008.

[5] A. Clauset, M. E. J. Newman, and C. Moore
Finding community structure in very large networks.
Physical Review E, **70**, 6 (Dec. 2004), 066111.

[6] N. Cristianini and J. Shawe-Taylor
An introduction to support vector machines.
Cambridge University Press, 2000.

[7] I. Derenyi, G. Palla, and T. Vicsek
Clique percolation in random networks.
Physical Review Letters, **94**, 16 (Apr. 2005), 160202.

[8] S. Fortunato
Community detection in graphs.
Physics Reports, **486**, 3–5 (Feb. 2010), 75–174.

[9] S. Fortunato and M. Barthélemy
Resolution limit in community detection.
Proceedings of the National Academy of Sciences, **104**, 1 (2007), 36.

[10] A. Galstyan and P. R. Cohen
Identifying covert sub-networks through iterative node classification.
In *Proceedings of the First International Conference on Intelligence Analysis*, 2005.

[11] M. Girvan and M. E. J. Newman
Community structure in social and biological networks.
Proceedings of the National Academy of Sciences of the United States of America, **99**, 12 (June 2002), 7821–7826.

[12] S. Gregory
An algorithm to find overlapping community structure in networks.
In *Knowledge Discovery in Databases: PKDD 2007*, 2007, 91–102.

[13] S. Gregory
Finding overlapping communities in networks by label propagation.
New Journal of Physics, **12** (2010), 103018.

[14] R. Guimerà and L. A. Nunes Amaral
Functional cartography of complex metabolic networks.
Nature, **433**, 7028 (Feb. 2005), 895–900.

[15] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral
Modularity from fluctuations in random graphs and complex networks.
Physical Review E, **70**, 2 (2004), 025101.

[16] J. Kubica, A. Moore, J. Schneider, and Y. Yang
Stochastic link and group detection.
In *Proceedings of the national conference on Artificial Intelligence*, 2002, 798–806.

[17] A. Lancichinetti and S. Fortunato
Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities.
Physical Review E, **80**, 1 (2009), 016118.

[18] A. Lancichinetti and S. Fortunato
Community detection algorithms: A comparative analysis.
Physical Review E, **80**, 5 (Nov. 2009), 056117.

[19] A. Lancichinetti, S. Fortunato, and J. Kertész
Detecting the overlapping and hierarchical community structure of complex networks.
New J. Phys., **11** (2009), 033015.

[20] J. Moody
Race, School integration, and friendship segregation in America.
American Journal of Sociology, **107**, 3 (Nov. 2001), 679–716.

[21] M. E. J. Newman
The structure and function of complex networks.
SIAM Review, **45**, 2 (Jan. 2003), 167–256.

[22] M. E. J. Newman
Analysis of weighted networks.
Physical Review E, **70**, 5 (Nov. 2004), 056131.

[23] M. E. J. Newman and M. Girvan
Finding and evaluating community structure in networks.
Physical Review E, **69**, 2 (Feb. 2004), 026113.

[24] R. K. Pan and S. Sinha
Modularity produces small-world networks with dynamical time-scale separation.
EPL (Europhysics Letters), **85** (Mar. 2009), 68006.

[25] U. N. Raghavan, R. Albert, and S. Kumara
Near linear time algorithm to detect community structures in large-scale networks.
Physical Review E, **76**, 3 (2007), 036106.

[26] J. Reichardt and S. Bornholdt
Clustering of sparse data via network communities—A prototype study of a large online market.
Journal of Statistical Mechanics: Theory and Experiment, (June 2007), P06016.

[27] M. Rosvall and C. T. Bergstrom
Maps of random walks on complex networks reveal community structure.
Proceedings of the National Academy of Sciences, **105**, 4 (Jan. 2008), 1118–1123.

[28] T. Teitelbaum, P. Balenzuela, P. Cano, and J. M. Buldú
Community structures and role detection in music networks.
Chaos: An Interdisciplinary Journal of Nonlinear Science, **18** (2008), 043105.

[29] S. Valverde, G. Theraulaz, J. Gautrais, V. Fourcassie, and R. V. Sole
Self-organization patterns in wasp and open source communities.
Intelligent Systems, IEEE, **21**, 2 (2006), 36–40.

[30] K. Wakita and T. Tsurumi
Finding community structure in mega-scale social networks.
In *Proceedings of the 16th International Conference on World Wide Web*, Banff, Alberta, Canada, 2007, 1275–1276.

[31] D. J. Watts and S. H. Strogatz
Collective dynamics of 'small-world' networks.
Nature, **393**, 6684 (June 1998), 440–442.

Leto Peel received a M.Eng. in knowledge engineering from the Engineering Mathematics Department of the University of Bristol in 2006.

Since 2006 he has been a research scientist for BAE Systems Advanced Technology Centre in Bristol. He worked in the area of Integrated Vehicle Health Management for 3 years and in 2008 was the winner in the IEEE GOLD category of the Prognostics and Health Management Data Challenge. In 2009 he started an Engineering Doctorate at the Eng.D. VEIV Centre for Doctoral Training at University College London and is supervised by Dr. Simon Julier and Professor John Shawe-Taylor. His research interests include a wide range of machine learning topics, in particular learning from relational data such as social networks.

Heterogeneous Track-to-Track Fusion

TING YUAN
YAAKOV BAR-SHALOM
XIN TIAN

Track-to-track fusion using estimates from multiple sensors can achieve better estimation performance than single sensor tracking. If the local sensors use different system models in different state spaces, the problem of heterogeneous track-to-track fusion arises. Compared with homogeneous track-to-track fusion that assumes the same system model for different sensors, the heterogeneous case poses two major challenges. The first one is that we have to fuse estimates from different state spaces (related by a certain nonlinear transformation). The second is the estimation errors' dependence problem, which is generally recognized as the "common process noise effect." Different heterogeneous track-to-track fusion approaches, namely, the linear minimum mean square error approach and the maximum likelihood approach, are presented and compared with the corresponding centralized measurement tracker/fuser (also known as measurement-to-track fuser).

Manuscript submitted on February 24, 2011; revised June 20, 2011; released for publication July 13, 2011.

Refereeing of this contribution was handled by Dr. Peter K. Willett.

Authors' address: University of Connecticut, ECE Department, 371 Fairfield Way, Unit 2157, Storrs, CT 06279, E-mail: ({tintti, ybs, xin.tian}@engr.uconn.edu).

1. INTRODUCTION

In a multisensor tracking system, the best target state estimation performance is obtained by a centralized tracker/fuser (CTF), by directly sending to the fusion center (FC) all the measurements of the local sensors.[1] However, in many practical situations, because of communication constraints, each local sensor has its own information processing system and sends only tracks to the FC, which appropriately fuses tracks from different local sensors to achieve comparable estimation performance to that of the CTF [3].

For track-to-track fusion (T2TF) from homogeneous local trackers (which use the same target state space), the "common process noise effect" (quantified by the crosscovariance matrix) has been theoretically well-established [3]. However, there is no known way for the calculation of the crosscovariance matrix in the case of heterogeneous local trackers (which use the different target state spaces). The difficulty to evaluate the crosscovariance matrix in the heterogeneous case is that it requires to capture the "common" part of process noises from different state spaces to quantify the crosscorrelation.

In the literature there are few works dealing with the model heterogeneity. A heterogeneous T2TF fusion approach was presented in [6] to fuse the tracks from an active sensor and a passive sensor with different state vectors. However, the fusion was done by using the full Cartesian state estimates (from an active sensor) to update the smaller angular state estimates (from a passive sensor). An expression for the steady state crosscovariance matrix for dissimilar sensors (of the same state vector but with different measurement noise variances) employing α-β filters was derived in [9]. For this specified case, a condition to guarantee the positivity of crosscovariance matrix was presented, which does not always hold in the heterogeneous case.

The goal of this paper is to fuse the tracks from heterogeneous local sensors (an active and a passive one) with different state spaces to yield fused estimates in the full state space and evaluate the performance of the resulting heterogeneous T2TF. The fusion configuration considered is the one without memory at the FC and no feedback to the local sensors (T2TFwoMnf in the terminology of [10]).

In view of the fact that there is no known way to evaluate the crosscovariance of the estimation errors in the case of heterogeneous sensors, a Monte Carlo (MC) investigation of these errors' crosscorrelations is carried out.

The paper is organized as follows. Section 2 formulates the heterogenous T2TF problem. Two approaches, namely, the linear minimum mean square error (LMMSE) and maximum likelihood (ML) heterogeneous T2TF are presented in Section 3. The cross-

[1]The superiority of CTF over other configurations can be proved only for the linear case [3].

correlation analysis by MC simulations is presented in Section 4. Section 5 evaluates the proposed approaches in a tracking scenario with an active sensor and a passive sensor. Section 6 provides conclusions.

2. THE HETEROGENOUS TRACK-TO-TRACK FUSION PROBLEM

Without loss of generality, consider the following state-space models

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

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

at sensor i and

$$\mathbf{x}^j(k+1) = f^j(\mathbf{x}^j(k)) + \mathbf{v}^j(k) \tag{3}$$

$$\mathbf{z}^j(k) = h^j(\mathbf{x}^j(k)) + \mathbf{w}^j(k) \tag{4}$$

at sensor j. In the above, $f^s[\cdot]$ and $h^s[\cdot]$, $s = i, j$, are different and can be nonlinear; $\mathbf{v}^s(\cdot)$ and $\mathbf{w}^s(\cdot)$, $s = i, j$, are the process and measurement noises, respectively.

Further, note that $\mathbf{x}^i$ and $\mathbf{x}^j$ are in different state spaces. Let $\mathbf{x}^i$ be the larger dimension state (e.g., full Cartesian position and velocity in two-dimensional space for tracking with an active sensor)

$$\mathbf{x}^i = [x \quad \dot{x} \quad y \quad \dot{y}]' \tag{5}$$

and $\mathbf{x}^j$ be the smaller dimension state (e.g., angular position and velocity for tracking with a passive sensor)

$$\mathbf{x}^j = [\theta \quad \dot{\theta}]'. \tag{6}$$

These state vectors have the nonlinear relationship

$$\mathbf{x}^j \triangleq g(\mathbf{x}^i). \tag{7}$$

The two sensors are assumed synchronized[2] and the time index k for sampling time t_k will be omitted if there is no ambiguity.

The corresponding estimates (approximate conditional means) at these heterogeneous local sensors are $\hat{\mathbf{x}}^i$ with (conditional) covariance matrix

$$P^i \triangleq E[(\mathbf{x}^i - \hat{\mathbf{x}}^i)(\mathbf{x}^i - \hat{\mathbf{x}}^i)'] \tag{8}$$

and $\hat{\mathbf{x}}^j$ with (conditional) covariance matrix

$$P^j \triangleq E[(\mathbf{x}^j - \hat{\mathbf{x}}^j)(\mathbf{x}^j - \hat{\mathbf{x}}^j)']. \tag{9}$$

The problem is how to carry out the fusion of the estimate $\hat{\mathbf{x}}^i$ with P^i and the estimate $\hat{\mathbf{x}}^j$ with P^j to achieve a better estimation performance for the full state of interest $\mathbf{x}^i$.

[2]Generalization to asynchronous sensors is possible [11], but the notations become very cumbersome. Without considering the crosscovariance matrix, the extension to asynchronous case is straightforward. If the crosscovariance matrix is considered (for the configuration with no memory at the FC and no information feedback to the local trackers), each track's latest estimate available at the FC is predicted to the fusion time and then they are fused using the corresponding covariance matrices.

3. THE HETEROGENOUS TRACK-TO-TRACK FUSION

To illustrate the effect of the crosscovariance, consider the simple homogeneous T2TF in the linear-Gaussian and symmetric case with the local track covariance matrices $P_S^1 = P_S^2 = P_S$ and the crosscovariance matrices $P_S^{12} = P_S^{21} = P_S^X$. The resulting fused estimate and its covariance matrix are [3]

$$\hat{\mathbf{x}}_S^F = \tfrac{1}{2}(\hat{\mathbf{x}}_S^1 + \hat{\mathbf{x}}_S^2) \tag{10}$$

$$P_S^F = P_S^1 - (P_S^2 - P_S^{12})(P_S^1 + P_S^2 - P_S^{12} - P_S^{21})(P_S^1 - P_S^{21}) \tag{11}$$

$$= \tfrac{1}{2}P_S + \tfrac{1}{2}P_S^X. \tag{12}$$

In this case, the fused estimate $\hat{\mathbf{x}}_S^F$ in (10) is independent of the crosscovariance because of the assumed symmetry. However, the corresponding covariance P_S^F in (11) has a term that depends on the crosscovariance. If $P_S^X > 0$, the fusion is optimistic if one ignores the crosscovariance (in which case the fuser calculated covariance is $\frac{1}{2}P_S$, i.e., smaller than what it should be); if $P_S^X < 0$, the fusion is pessimistic.

The crosscovariance for homogeneous fusion follows from a Lyapunov equation [3] and, consequently, it is always positive semi-definite. In the heterogeneous case while there is no known way to compute the crosscovariance matrix, shown in Appendix C using MC simulations, some of the crosscorrelations are positive and some are negative. They depend on the relative geometry of the two sensors and the target, as well as the target maneuvers. To further complicate the situation, the maneuvers are unknown deterministic, rather than (zero-mean white) process noise and the crosscovariance based on the process noise can be substantially different from what the maneuver causes.

The following subsections present two fusers that assume the crosscovariance is available.

3.1. The LMMSE Fuser

The first approach to heterogeneous T2TF is to use the linear technique based on the fundamental equations of LMMSE estimation [2]. Considering the full state estimate $\hat{\mathbf{x}}^i$ as the prediction and the smaller state estimate $\hat{\mathbf{x}}^j$ as the measurement, we have the LMMSE fused estimate

$$\hat{\mathbf{x}}^i_{\text{LMMSE}} = \hat{\mathbf{x}}^i + P_{\mathbf{xz}}P_{\mathbf{zz}}^{-1}[\hat{\mathbf{x}}^j - g(\hat{\mathbf{x}}^i)] \tag{13}$$

with the corresponding fused covariance matrix

$$P^i_{\text{LMMSE}} = P^i - P_{\mathbf{xz}}P_{\mathbf{zz}}^{-1}P_{\mathbf{xz}}' \tag{14}$$

where (as the details shown in Appendix A)

$$P_{\mathbf{xz}} \triangleq E[(\mathbf{x}^i - \hat{\mathbf{x}}^i)(\hat{\mathbf{x}}^j - g(\hat{\mathbf{x}}^i))'] \approx P^i(G^i)' - P^{ij} \tag{15}$$

$$P_{\mathbf{zz}} \triangleq E[(\hat{\mathbf{x}}^j - g(\hat{\mathbf{x}}^i))(\hat{\mathbf{x}}^j - g(\hat{\mathbf{x}}^i))'] \approx P^j - G^iP^{ij} - P^{ji}(G^i)' + G^iP^i(G^i)' \tag{16}$$

with G^i the Jacobian of $g(\mathbf{x}^i)$

$$G^i \triangleq [\nabla_{\mathbf{x}^i} g(\mathbf{x}^i)']'_{\mathbf{x}^i=\hat{\mathbf{x}}^i} \tag{17}$$

and P^{ij} the crosscovariance matrix

$$P^{ij} \triangleq E[(\mathbf{x}^i - \hat{\mathbf{x}}^i)(\mathbf{x}^j - \hat{\mathbf{x}}^j)']. \tag{18}$$

3.2. The ML Fuser

Under the Gaussian assumption, the heterogeneous T2TF problem can be solved by minimizing the negative log-likelihood function[3]

$$\begin{aligned} L &= -\ln p(\hat{\mathbf{x}}^i, \hat{\mathbf{x}}^j | \mathbf{x}^i) \\ &\propto \left(\begin{bmatrix} \hat{\mathbf{x}}^i \\ \hat{\mathbf{x}}^j \end{bmatrix} - \begin{bmatrix} \mathbf{x}^i \\ \mathbf{x}^j \end{bmatrix} \right)' P^{-1} \left(\begin{bmatrix} \hat{\mathbf{x}}^i \\ \hat{\mathbf{x}}^j \end{bmatrix} - \begin{bmatrix} \mathbf{x}^i \\ \mathbf{x}^j \end{bmatrix} \right) \end{aligned} \tag{19}$$

where (7) has been used and

$$P = \begin{bmatrix} P^i & P^{ij} \\ P^{ji} & P^j \end{bmatrix}. \tag{20}$$

Then the ML fused estimate is the solution of

$$\nabla_{\mathbf{x}^i} L = 0. \tag{21}$$

Because of the nonlinearity of the function $g(\mathbf{x}^i)$, there is no explicit expression for the solution of (21). It can be solved by a numerical search, e.g., the gradient projection algorithm. The result is denoted as $\hat{\mathbf{x}}^i_{\mathrm{ML}}$ and the corresponding covariance matrix is

$$P^i_{\mathrm{ML}} = \left([I \quad G^i] P^{-1} \begin{bmatrix} I \\ G^i \end{bmatrix} \right)^{-1} \tag{22}$$

where G^i is defined in (17) and I is the identity matrix (4×4 in our case).

The results of $\hat{\mathbf{x}}^i_{\mathrm{LMMSE}}$ with P^i_{LMMSE} and $\hat{\mathbf{x}}^i_{\mathrm{ML}}$ with P^i_{ML} will be examined and compared with the CTF which processes all the measurements (from both the active and the passive sensor) in the FC in the simulation section.

4. THE CROSSCORRELATION IN HETEROGENEOUS FUSION

It has been recognized that, although different local sensors typically have independent measurement noises, the process noises for the motion models at these sensors belong to the same target and, consequently, will lead to (cross)correlated state estimation errors. This is the so-called "common process noise effect" [3]. For the heterogenous case, the common process noise effect, as it is embedded into the estimates from different sensors for the same target, also exists. However, since the estimates are in different state spaces, there is no known way to capture the "common" part exactly.

The dependence of the estimation errors can be quantified by the crosscovariance matrix, and the more accurately the crosscovariance matrix is obtained, the better the heterogeneous track-to-track fusion performance will be. However, the difference between the motion models for different sensors prohibits the evaluation of the crosscovariance matrix by the exact method described in [3] (limited to the homogeneous case and linear systems). Even this exact method is not considered practical since it requires information that is typically not available at the FC (the local filter gains).

While process noise is used in the motion equations to model the target maneuvers,[4] these maneuvers are, however, not stochastic processes. Consequently, MC simulations will be used to evaluate the crosscorrelation of the estimation errors from different sensors. As shown in Appendix C, considering the estimates from different local sensors in each MC run as one sample and evaluating the sample crosscorrelation coefficients, we observe both negative and positive crosscorrelations of the estimation errors from the heterogeneous local sensors in different parts of the target trajectory.

The fact that these crosscorrelations can be, unlike in the linear homogeneous case (when they are always positive), sometimes positive and sometimes negative is shown as follows. According to (14), the information matrix (assuming P^{ij} is available) is

$$\begin{aligned} J &= (P^F)^{-1} \\ &= [P^i - [P^i(G^i)' - P^{ij}][P^j - G^i P^{ij} - P^{ji}(G^i)' + G^i P^i (G^i)']^{-1} \\ &\quad \times [P^i(G^i)' - P^{ij}]']^{-1} \\ &\triangleq [P^i - [P^i(G^i)' - P^{ij}][P^j + G^i P^i (G^i)' + U]^{-1} \\ &\quad \times [P^i(G^i)' - P^{ij}]']^{-1} \end{aligned} \tag{23}$$

where

$$U \triangleq -G^i P^{ij} - P^{ji}(G^i)'. \tag{24}$$

Assuming $P^{ij} = 0$ (its elements are all zero), designated as the "uncorrelated" assumption (denoted concisely as "uncorr"), then (23) can be simplified (by the matrix inversion lemma) as

$$\begin{aligned} J(P^{ij} = 0) &= [P^i - P^i(G^i)'[P^j + G^i P^i (G^i)']^{-1} G^i (P^i)']^{-1} \\ &= (P^i)^{-1} - [(P^i)^{-1} P^i (G^i)'] \\ &\quad \times [G^i (P^i)' (P^i)^{-1} P^i (G^i)' - P^j - G^i P^i (G^i)']^{-1} \\ &\quad \times [(P^i)^{-1} P^i (G^i)']' \\ &= (P^i)^{-1} + (G^i)'(P^j)^{-1} G^i. \end{aligned} \tag{25}$$

[3] As it is pointed out in [4], the LMMSE T2TF approach is, in the linear Gaussian case, optimal in ML sense.

[4] The whiteness is necessary so the state is a Markov process, a sine qua non prerequisite for any recursive estimation algorithm [2].

If $P^{ij} \neq 0$ (this is denoted as "corr" for conciseness), then we have

$$\begin{aligned} J(P^{ij} \neq 0) &= (P^i)^{-1} - [(G^i)' - (P^i)^{-1}P^{ij}] \\ &\quad \cdot [[P^i(G^i)' - P^{ij}]'(P^i)^{-1}[P^i(G^i)' - P^{ij}] \\ &\quad - [P^j + G^iP^i(G^i)' + U]]^{-1} \\ &\quad \cdot [(G^i)' - (P^i)^{-1}P^{ij}]' \\ &\triangleq (P^i)^{-1} + [(G^i)' - (P^i)^{-1}P^{ij}] \\ &\quad \cdot [P^j + W]^{-1}[(G^i)' - (P^i)^{-1}P^{ij}]' \end{aligned} \tag{26}$$

where

$$\begin{aligned} W &\triangleq [G^iP^i(G^i)' + U] - [P^i(G^i)' - P^{ij}]'(P^i)^{-1}[P^i(G^i)' - P^{ij}] \\ &= -P^{ji}(P^i)^{-1}P^{ij}. \end{aligned} \tag{27}$$

Equation (26) can be written as (the generic matrix inversion lemma is used)

$$\begin{aligned} J(P^{ij} \neq 0) &= (P^i)^{-1} + [(G^i)' - (P^i)^{-1}P^{ij}] \\ &\quad \cdot [(P^j)^{-1} - (P^j)^{-1}W(P^j)^{-1}[I + W(P^j)^{-1}]^{-1}] \\ &\quad \cdot [(G^i)' - (P^i)^{-1}P^{ij}]' \\ &\triangleq (P^i)^{-1} + [(G^i)' - (P^i)^{-1}P^{ij}][(P^j)^{-1} - K] \\ &\quad \cdot [(G^i)' - (P^i)^{-1}P^{ij}]' \\ &\triangleq (P^i)^{-1} + (G^i)'(P^j)^{-1}G^i - K_c \\ &= J(P^{ij} = 0) - K_c \end{aligned} \tag{28}$$

where I is the identity matrix and

$$K \triangleq (P^j)^{-1}W(P^j)^{-1}[I + W(P^j)^{-1}]^{-1} \tag{29}$$

$$\begin{aligned} K_c &\triangleq [(G^i)' - (P^i)^{-1}P^{ij}]K[(G^i)' - (P^i)^{-1}P^{ij}]' \\ &\quad + (P^i)^{-1}P^{ij}(P^j)^{-1}G^i + [(P^i)^{-1}P^{ij}(P^j)^{-1}(G^i)]' \\ &\quad - [(P^i)^{-1}P^{ij}](P^j)^{-1}[(P^i)^{-1}P^{ij}]'. \end{aligned} \tag{30}$$

Setting $P^{ij} = 0$ and $P^{ij} \neq 0$ for the estimation from the heterogenous local sensors correspond to "uncorr" and "corr" assumptions, respectively. For the homogeneous case, the crosscovariance matrix is always positive; this follows from the Lyapunov equation (9.3.2–3) in [3]. However, in the heterogeneous (and nonlinear) case, K_c may be indefinite (some eigenvalues are positive and some negative). Therefore, accounting for it (as opposed to assuming it zero) yields the exact (optimal) variance larger in some state components and smaller in others.

The results using a simple functional model of the crosscorrelation of the estimation errors ($\tilde{\mathbf{x}}^i$ and $\tilde{\mathbf{x}}^j$), based on the polar-to-Cartesian transformation, are shown in Appendix D and they do not provide any perceivable benefits. As discussed above, for the linear-Gaussian and symmetric case, neglecting the common process noise makes the fusion optimistic. For the nonlinear case examined, neglecting it makes the fusion sometimes optimistic and sometimes pessimistic, but the effect is small. This supports the approach of ignoring the dependency between the tracks from different local sensors.[5] Thus, since the maneuvers are unknown and scenario dependent, we pursue the heterogeneous T2TF without considering the crosscorrelation between the estimation errors.

5. SIMULATION RESULTS

A typical two-dimensional scenario for heterogenous T2TF is with an active sensor located at (x_a, y_a), with measurements of target range and azimuth angle

$$r = \sqrt{(x - x_a)^2 + (y - y_a)^2} + w_r \tag{31}$$

$$\theta_a = \tan^{-1}\left(\frac{y - y_a}{x - x_a}\right) + w_a \tag{32}$$

and a passive sensor located at (x_p, y_p), with the azimuth angle measurements

$$\theta_p = \tan^{-1}\left(\frac{y - y_p}{x - x_p}\right) + w_p \tag{33}$$

where w_r, w_a and w_p are mutually independent zero mean white Gaussian noises with standard deviations (SD) σ_r, σ_a and σ_p, respectively.

The ground truth is a target moving with a constant speed of 250 m/s with initial state in Cartesian coordinates (with position in m)

$$\begin{aligned} \mathbf{x}^i(0) &= [x(0) \quad \dot{x}(0) \quad y(0) \quad \dot{y}(0)]' \\ &= [0 \quad 0 \quad 20000 \quad 250]'. \end{aligned} \tag{34}$$

At $t = 100$ s it starts a left turn of 2°/s (about 30 mrad/s) for 30 s, then continues straight until $t = 200$ s, at which time it turns right with 1°/s for 45 s, then left with 1°/s for 90 s, then right with 1°/s for 45 s, then continues straight until 50 s.

The measurements of the active sensor located at $(-6 \cdot 10^4, 2 \cdot 10^4)$ m are made every $T_a = 5$ s, starting from $k = 0$ with measurement noise SD as $\sigma_r = 20$ m and $\sigma_a = 5$ mrad. An unbiased measurement conversion from polar coordinates to Cartesian coordinates (denoted as $\mathbf{z}^i$ with $\mathbf{w}^i$ for polar coordinates and $\mathbf{z}_c^i$ with $\mathbf{w}_c^i$ for Cartesian coordinates, respectively) is done for the active sensor measurements for filtering [2]. The measurements of the passive sensor located at $(-5 \cdot 10^4, 4 \cdot 10^4)$ m are made every $T_p = 1$ s, starting from $k = 0$ with measurement noise $\sigma_p = 1$ mrad. This scenario is shown in Fig. 1.

The active sensor uses an interacting multiple model (IMM) estimator using continuous time white noise

[5] We are grateful for the anonymous reviewer who summarized so well our findings.

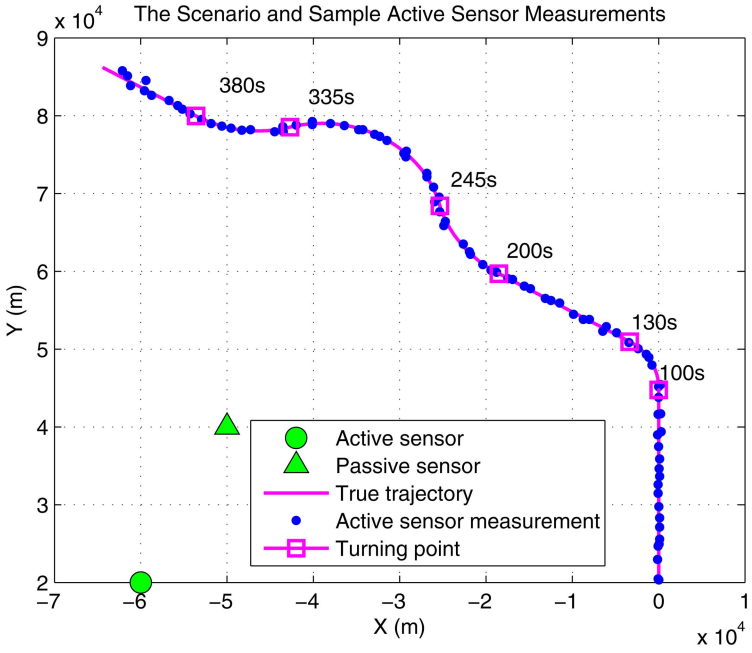


Fig. 1. The scenario.

acceleration (CWNA) model [2]. The passive sensor uses, for reasons shown in Appendix E, a linear Kalman filter (KF) using a continuous time Wiener process acceleration (CWPA) model [2].

5.1. The Active Sensor IMM Estimator Design

For the active sensor IMM, in order to cover the uniform motion segments and maneuvering segments in the trajectory, two modes are used: one mode using a linear nearly constant velocity (NCV) model that includes 4 state components (the Cartesian position and velocity in X- and Y-coordinates) and the other mode using a nearly coordinated turn (NCT) model that includes that 5 state components (the Cartesian position and velocity in X- and Y-coordinates and the turn rate Ω)

$$\begin{aligned}\mathbf{x}_a^i(k) &\triangleq [x(k) \quad \dot{x}(k) \quad y(k) \quad \dot{y}(k) \quad \Omega(k)]' \\ &= [\mathbf{x}^i(k)' \quad \Omega(k)]'. \end{aligned} \tag{35}$$

Since the conventional mixing [2] in an IMM estimator with modes that have unequal dimension state vectors will lead to bias in the estimates of the non-common state component(s), the unbiased mixing approach described in [13] is used to overcome this bias problem (for the turn rate in this case).

The NCV model for uniform motion segment is implemented as the following discretized CWNA model [2] (with low-level process noise)

$$\mathbf{x}^i(k+1) = F^i(k)\mathbf{x}^i(k) + \mathbf{v}^i(k) \tag{36}$$

$$\mathbf{z}_c^i(k) = H^i(k)\mathbf{x}^i(k) + \mathbf{w}_c^i(k) \tag{37}$$

where, with T_a is the sampling interval of the active sensor,[6]

$$F^i = \begin{bmatrix} 1 & T_a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_a \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{38}$$

$$H^i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{39}$$

and the covariance matrix of the process noise $\mathbf{v}^i(k)$ is

$$\begin{aligned} Q^i(k) &\triangleq E[\mathbf{v}^i(k)\mathbf{v}^i(k)'] \\ &= \begin{bmatrix} \begin{bmatrix} \frac{T_a^3}{3} & \frac{T_a^2}{2} \\ \frac{T_a^2}{2} & T_a \end{bmatrix} q_a & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \begin{bmatrix} \frac{T_a^3}{3} & \frac{T_a^2}{2} \\ \frac{T_a^2}{2} & T_a \end{bmatrix} q_a \end{bmatrix} \end{aligned} \tag{40}$$

where the continuous time process noise "intensities" q_a is the power spectral density (PSD). Note that the process noise induced root mean square (RMS) change in the velocity over a sampling interval T_a is

$$d_a \triangleq \frac{\sqrt{q_a T_a}}{T_a} \tag{41}$$

whose physical dimensions is linear acceleration [12].

[6] Different sampling rates may be possible and T_a is replaced by the time-variant $T_a(k) \triangleq t(k+1) - t(k)$.

The NCT model, commonly refers to a target maneuver executed under nearly constant speed along a nearly circular path, is implemented as the following discretized continuous-time coordinated turn dynamic model [8]

$$\mathbf{x}_a^i(k+1) = f_a^i[\mathbf{x}_a^i(k)] + \mathbf{v}_a^i[\mathbf{x}_a^i(k)] \tag{42}$$

$$\mathbf{z}_c^i(k) = H_a^i \mathbf{x}_a^i(k) + \mathbf{w}_c^i(k) \tag{43}$$

where

$$f_a^i[\mathbf{x}_a^i(k)] = \begin{bmatrix} x(k) + T_a\dot{x}(k) - T_a^2\Omega(k)\dot{y}(k)/2 \\ \dot{x}(k) - T_a\Omega(k)\dot{y}(k) - T_a^2\Omega(k)^2\dot{x}(k)/2 \\ y(k) + T_a\dot{y}(k) + T_a^2\Omega(k)\dot{x}(k)/2 \\ \dot{y}(k) + T_a\Omega(k)\dot{x}(k) - T_a^2\Omega(k)^2\dot{y}(k)/2 \\ \Omega(k) \end{bmatrix} \tag{44}$$

$$H_a^i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{45}$$

and the process noise for the NCT model depends on current target state (target velocity components and turn rate). The covariance matrix of the target state-dependent process noise $\mathbf{v}_a^i[\mathbf{x}_a^i(k)]$ can be roughly shown as (the detailed form can be found in Appendix B)[7]

$$Q_a^i[\mathbf{x}_a^i(k)] \triangleq E\{\mathbf{v}_a^i[\mathbf{x}_a^i(k)]\mathbf{v}_a^i[\mathbf{x}_a^i(k)]'\} = \begin{bmatrix} \frac{T_a^3}{3}\frac{\dot{x}(k)^2}{\dot{x}(k)^2+\dot{y}(k)^2}q_a & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \frac{T_a^3}{3}\frac{\dot{y}(k)^2}{\dot{x}(k)^2+\dot{y}(k)^2}q_a & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & Tq_\Omega \end{bmatrix} \tag{46}$$

where the continuous time process noise "intensities" q_a and q_Ω are the PSDs.

Note that the process noise induced RMS change over a sampling interval T_a in the velocity is as in (41) and in the turn rate is

$$d_\Omega \triangleq \frac{\sqrt{q_\Omega T_a}}{T_a} \tag{47}$$

whose physical dimensions is turn acceleration [12]. The d_a and d_Ω are the design values used to select the process noise PSDs. A guideline for the choice of these process noise intensities for the NCT model is shown in Appendix B.

As the NCT model described in (42) is nonlinear, extended KF (EKF) has been used as the mode-matched filter for the NCT model in active sensor IMM. We only use the estimate $\hat{\mathbf{x}}^i(k)$ (from $\hat{\mathbf{x}}_a^i(k)$) and the corresponding covariance matrix $P^i(k)$ for the fusion.

TABLE I
The RMS Change Rate Due to Process Noise

	d_a (m/s^2)	d_Ω (mrad/s^2)
Mode 1 (NCV)	0.2	N/A
Mode 2 (NCT)	1	2

For the active sensor IMM estimator with unbiased converted measurements (in Cartesian coordinates), the process noises design values are summarized in Table I and the corresponding transition probability matrix is (based on the mean sojourn time [2])

$$\pi_{CT} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix} \tag{48}$$

with initial mode probability vector $[0.9, 0.1]$.

5.2 The Passive Sensor KF Estimator Design

For the passive sensor, as pointed out in Appendix C, the target maneuvering index is very small and the target maneuvers are nearly unobservable by the passive sensor. Consequently, a single model filter (i.e., a linear KF) has been chosen as the local estimator, with the state vector

$$\mathbf{x}_p^j \triangleq [\theta \quad \dot{\theta} \quad \ddot{\theta}]' = [\mathbf{x}^j(k)' \quad \ddot{\theta}]'. \tag{49}$$

The discretized CWPA model [2] in the angle, angle rate and angle acceleration space is

$$\mathbf{x}_p^j(k+1) = F_p^j \mathbf{x}_p^j(k) + \mathbf{v}_p^j(k) \tag{50}$$

$$\mathbf{z}_p^j(k) = H_p^j \mathbf{x}_p^j(k) + \mathbf{w}_p^j(k) \tag{51}$$

where, with T_p the sampling interval of the passive sensor, we have

$$F_p^j = \begin{bmatrix} 1 & T_p & \frac{T_p^2}{2} \\ 0 & 1 & T_p \\ 0 & 0 & 1 \end{bmatrix} \tag{52}$$

$$H_p^j = [1 \quad 0 \quad 0] \tag{53}$$

[7]This "target state-dependent" process noise covariance matrix from [8] yielded superior RMSE performance (but no major change in the consistency) compared with the simplified covariance matrix from [2], Sec. 11.7.

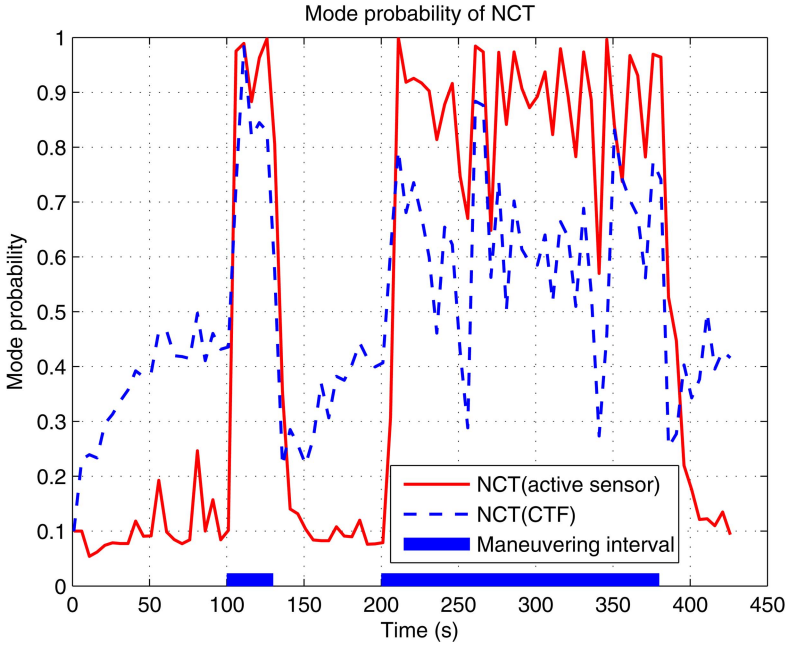


Fig. 2. The maneuvering mode probability (NCT) in the active sensor IMM and CTF IMM.

and the covariance matrix of the process noise is

$$Q_p^j(k) \triangleq E[\mathbf{v}_p^j(k)\mathbf{v}_p^j(k)'] = \begin{bmatrix} \frac{T_p^5}{20} & \frac{T_p^4}{8} & \frac{T_p^3}{6} \\ \frac{T_p^4}{8} & \frac{T_p^3}{3} & \frac{T_p^2}{2} \\ \frac{T_p^3}{6} & \frac{T_p^2}{2} & T_p \end{bmatrix} q_p. \tag{54}$$

Note that for the PSD q_p, the process noise induced RMS change in the angular acceleration over T_p is

$$d_p \triangleq \frac{\sqrt{q_p T_p}}{T_p} \tag{55}$$

whose physical dimension is angular jerk (derivative of acceleration).

The process noise design value chosen for the passive sensor is $d_p = 0.04$ (mrad/s^3). We only use the estimate $\hat{\mathbf{x}}^j(k)$ (from $\hat{\mathbf{x}}_p^j(k)$) and the corresponding covariance matrix $P^j(k)$ for the fusion.

5.3. The Heterogeneous T2TF

The LMMSE and ML heterogeneous T2TF are carried out at the FC every $T_f = 5$ s under the "uncorr" assumption, with the local estimates $\hat{\mathbf{x}}^i(k)$ (from $\hat{\mathbf{x}}_a^i(k)$) and $\hat{\mathbf{x}}^j(k)$ (from $\hat{\mathbf{x}}_p^j(k)$) and their corresponding covariance matrices $P^i(k)$ and $P^j(k)$. The CTF uses the same IMM design (CTF IMM for short) as the active sensor IMM estimator. The FC can run the fusion at an arbitrarily low rate or "on demand."

1) *The LMMSE Fuser*: In Figs. 3 and 4, the root mean square errors (RMSE) for the LMMSE fuser (with $T_f = 5$ s under the "uncorr" assumption) are compared with those for the active sensor's IMM estimator and the CTF IMM in position and velocity, respectively. It can be seen that the LMMSE heterogeneous T2TF approach always provides significantly better estimation performance than the single (active) sensor case.

The LMMSE heterogeneous T2TF provides larger RMSE than the CTF IMM in the non-maneuvering intervals but smaller RMSE if there is a maneuver. This degradation of the CTF in both position and velocity during the maneuvering intervals is because the CTF is using an IMM estimator, which is inappropriate for the passive sensor (due to the very small maneuver index). While using the IMM estimator is generally beneficial for maneuvering targets, the use of an IMM estimator with a sensor that cannot "see" the maneuvers can lead to performance degradation (the CTF IMM's performance at some fusion points is even worse than the active sensor IMM's). As shown in Appendix E, the maneuvering index from the passive sensor's view is so small that when the passive sensor measurements (with higher sampling rate than those of the active sensor) are sent to FC and processed centrally, these measurements increase the uncertainty about the target maneuvers.

From the maneuvering mode probability (NCT) in the active sensor IMM and in the CTF IMM, shown in Fig. 2, it can be seen that the CTF IMM cannot "see" the maneuvers at the times when there is only a passive sensor measurement and its maneuvering mode probability becomes too small. The use of the passive measurements in the CTF IMM "clouds" the maneuvers—it is preferable to have an active sensor IMM (which does detect the maneuvers) and a passive sensor KF (since the passive sensor is almost "blind" to the maneuvers) and fuse the outputs of these two local trackers.

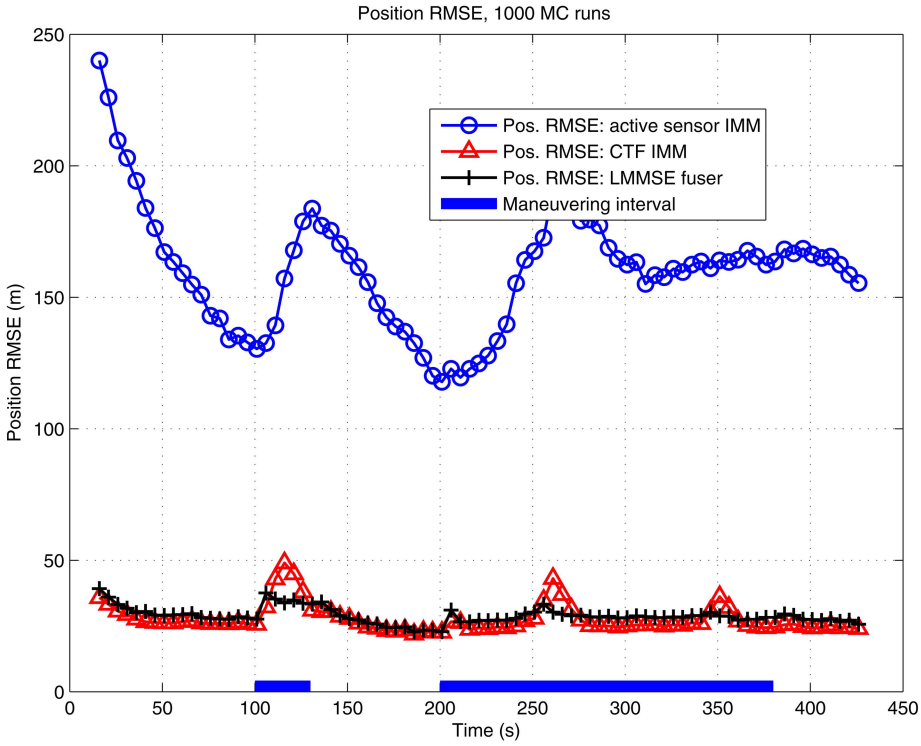


Fig. 3. The position RMSE for LMMSE fuser.

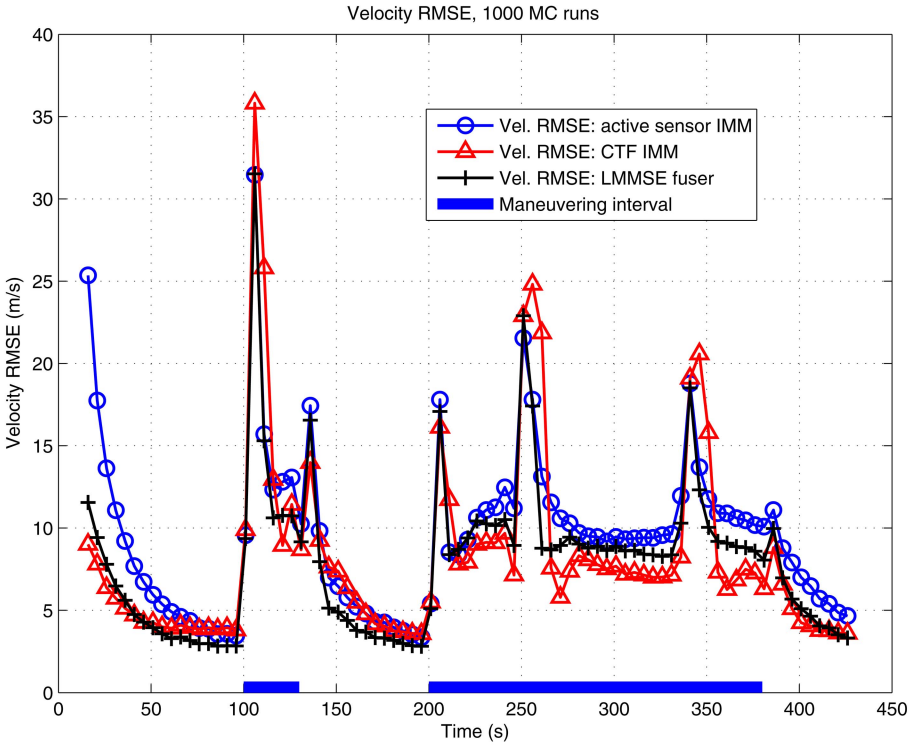


Fig. 4. The velocity RMSE for LMMSE fuser.

The observation from Figs. 3 and 4 that the CTF IMM performs during target maneuvers worse than the heterogeneous T2TF points out that the heterogeneous T2TF benefits from the freedom of having more suitable filters for the individual local sensors. This freedom can provide final fusion results comparable or even better than the corresponding CTF estimator.

We evaluate the fusion consistency of the LMMSE fuser by the normalized estimation error square (NEES) consistency test [2]. The NEES for the LMMSE fusion approach are shown in Fig. 5. The reason for the inconsistency of the fused estimates are (i) the local IMM estimator (for the active sensor) and the KF estimator (for the passive sensor) are not entirely

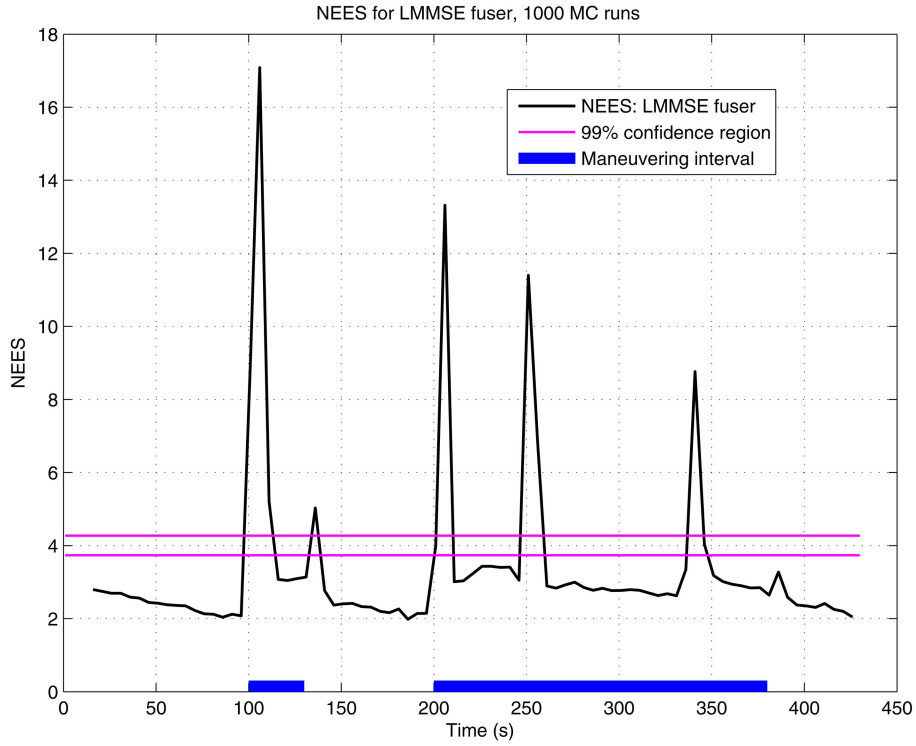


Fig. 5. The NEES for LMMSE fuser.

consistent[8] (as shown in Appendix F) and (ii) the cross-covariance has been assumed zero. Nevertheless, the quality of the estimates is improved by fusion, which justifies the approach. At this point, there is no known way to improve the sometimes optimistic, sometimes pessimistic behavior of the IMM—it is the inconsistency that drives its adaptation.

2) *The ML Fuser*: Using a numerical search (the gradient projection algorithm), the RMSE in position and velocity for the ML fuser are shown (with T_f = 5 s under the "uncorr" assumption) in Fig. 6 and Fig. 7, respectively. It can be seen that both the LMMSE fuser and the ML fuser give practically the same RMSE in position and velocity and both have better performances than the single (active) sensor case. As pointed out in [4], the LMMSE fuser is, in the linear-Gaussian case, actually optimal in the ML sense. Since the ML fuser in the heterogenous case (with nonlinearity) needs to be implemented by a time-consuming numerical search, the LMMSE fuser can be considered as an efficient and effective alternative for the ML fuser.

6. CONCLUSIONS

Examining the differences between the heterogeneous and homogeneous T2TF, this paper investigates the major difficulties of heterogenous T2TF. The LMMSE and the ML approaches for heterogenous T2TF are presented and compared with the corresponding CTF. The simulation study shows that both approaches can effectively achieve improved performance over the single sensor track quality and comparable performance to the CTF track. The use of the passive measurements in the CTF IMM "clouds" the maneuvers—it is preferable to have an active sensor IMM (which does detect the maneuvers) and a passive sensor KF (since the passive sensor is almost "blind" to the maneuvers) and fuse the outputs of these two local trackers. The freedom available to each local sensor to flexibly design a more suitable local estimator allows the heterogeneous T2TF approach to achieve a better estimation performance than the CTF IMM in the scenario considered. As the LMMSE T2TF has practically the same performance as the ML T2TF, it can be considered as an effective and efficient alternative for the numerical search required by the ML approach. The estimation errors' crosscorrelation has been examined by MC simulations. As it is impossible to predict maneuvers in a trajectory and there is no known way to correctly quantify the crosscorrelation of the estimation errors from heterogeneous local sensors, the heterogeneous T2TF was carried out assuming the tracks from the heterogeneous local sensors as uncorrelated.

[8]The IMM estimator is the worst estimator in terms of consistency except for all the other estimators [3]. However, it is the "short term" inconsistency that is the key for the capability of the IMM estimator to adapt itself to the observed behavior of the target (large innovations).

APPENDIX A. TAYLOR SERIES APPROXIMATION FOR THE LMMSE FUSER

By the first order Taylor expansion, we have

$$g(\mathbf{x}^i) \approx g(\hat{\mathbf{x}}^i) + G^i(\mathbf{x}^i - \hat{\mathbf{x}}^i) \tag{56}$$

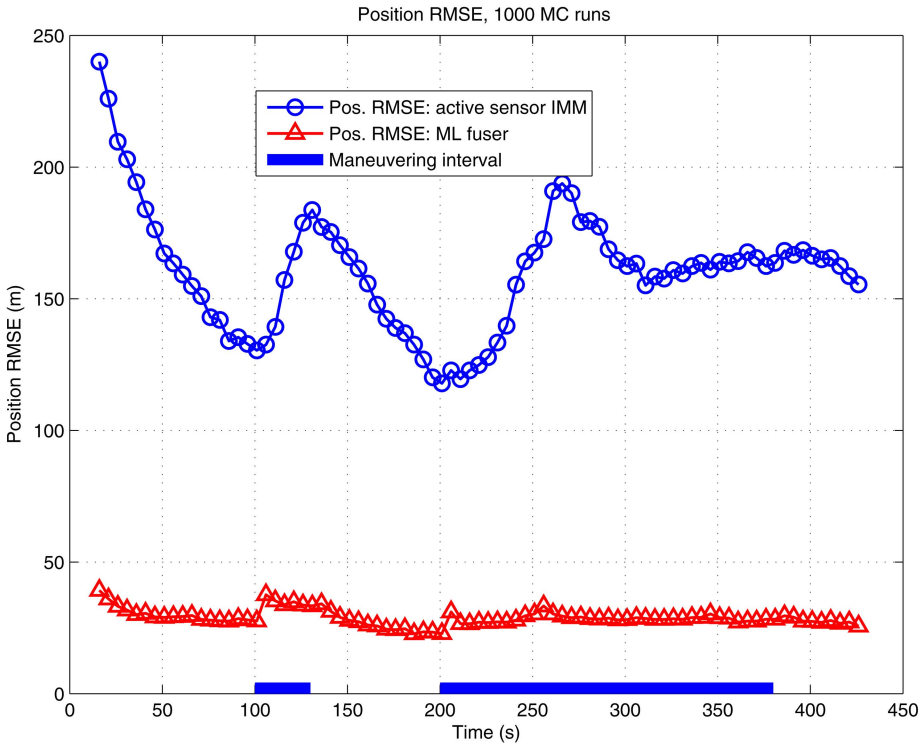


Fig. 6. The position RMSE for ML fuser.

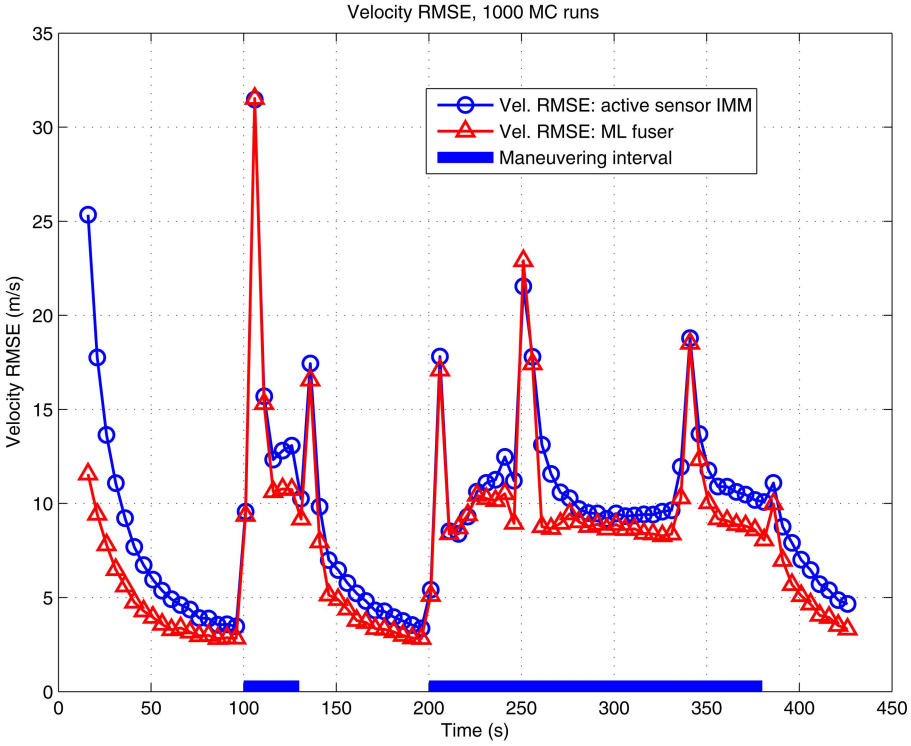


Fig. 7. The velocity RMSE for ML fuser.

where G^i is the Jacobian of $g(\mathbf{x}^i)$ evaluated at $\hat{\mathbf{x}}^i$, as defined in (17). Then (with the knowledge that $\mathbf{x}^j = g(\mathbf{x}^i)$)

$$\begin{aligned}\hat{\mathbf{x}}^j - g(\hat{\mathbf{x}}^i) &= \hat{\mathbf{x}}^j - g(\mathbf{x}^i) + G^i(\mathbf{x}^i - \hat{\mathbf{x}}^i) \\ &= G^i(\mathbf{x}^i - \hat{\mathbf{x}}^i) - (\mathbf{x}^j - \hat{\mathbf{x}}^j). \end{aligned} \tag{57}$$

So

$$\begin{aligned} P_{\mathbf{xz}} &\triangleq E[(\mathbf{x}^i - \hat{\mathbf{x}}^i)(\hat{\mathbf{x}}^j - g(\hat{\mathbf{x}}^i))'] \\ &\approx E[(\mathbf{x}^i - \hat{\mathbf{x}}^i)(G^i(\mathbf{x}^i - \hat{\mathbf{x}}^i) - (\mathbf{x}^j - \hat{\mathbf{x}}^j))'] \\ &= E[(\mathbf{x}^i - \hat{\mathbf{x}}^i)(\mathbf{x}^i - \hat{\mathbf{x}}^i)'](G^i)' - E[(\mathbf{x}^i - \hat{\mathbf{x}}^i)(\mathbf{x}^j - \hat{\mathbf{x}}^j)'] \\ &= P^i(G^i)' - P^{ij} \end{aligned} \tag{58}$$

and

$$P_{\mathbf{zz}} \triangleq E[(\hat{\mathbf{x}}^j - g(\hat{\mathbf{x}}^i))(\hat{\mathbf{x}}^j - g(\hat{\mathbf{x}}^i))']$$

$$\approx E[(G^i(\mathbf{x}^i - \hat{\mathbf{x}}^i) - (\mathbf{x}^j - \hat{\mathbf{x}}^j))(G^i(\mathbf{x}^i - \hat{\mathbf{x}}^i) - (\mathbf{x}^j - \hat{\mathbf{x}}^j))']$$

$$= P^j - G^i P^{ij} - P^{ji}(G^i)' + G^i P^i (G^i)'. \quad (59)$$

APPENDIX B. THE TARGET STATE-DEPENDENT PROCESS NOISE COVARIANCE MATRIX FOR THE NCT MODEL

As shown in [8], the target state-dependent process noise (in the Cartesian velocity model) is

$$\mathbf{v}_a^i[\mathbf{x}_a^i(k)] = D_a^i[\mathbf{x}_a^i(k)]\mathbf{v}_c^i(k) \quad (60)$$

where q_a and q_Ω are the continuous-time process noise PSDs and

$$s_1(k) = \frac{\dot{x}(k)}{\sqrt{\dot{x}^2(k) + \dot{y}^2(k)}} \quad (64)$$

$$s_2(k) = \frac{\dot{y}(k)}{\sqrt{\dot{x}^2(k) + \dot{y}^2(k)}} \quad (65)$$

$$s_3(k) = \frac{\dot{x}(k) - T_a\Omega(k)\dot{y}(k)}{\sqrt{\dot{x}^2(k) + \dot{y}^2(k)}} \quad (66)$$

$$s_4(k) = \frac{\dot{y}(k) + T_a\Omega(k)\dot{x}(k)}{\sqrt{\dot{x}^2(k) + \dot{y}^2(k)}}. \quad (67)$$

We have the corresponding target state-dependent process noise covariance matrix (the time index k is ignored for conciseness)

$$Q_a^i[\mathbf{x}_a^i(k)] \triangleq E\{\mathbf{v}_a^i[\mathbf{x}_a^i(k)]\mathbf{v}_a^i[\mathbf{x}_a^i(k)]'\} \quad (68)$$

$$= E\{[D_a^i[\mathbf{x}_a^i(k)]\mathbf{v}_c^i(k)][D_a^i[\mathbf{x}_a^i(k)]\mathbf{v}_c^i(k)]'\} \quad (69)$$

$$= \begin{bmatrix} \frac{T_a^3}{3}s_1^2 q_a & \frac{T_a^2}{2}s_1 s_3 q_a & \frac{T_a^3}{3}s_1 s_2 q_a & \frac{T_a^2}{2}s_1 s_4 q_a & 0 \\ \frac{T_a^2}{2}s_1 s_3 q_a & \frac{T_a^3}{3}\dot{y}^2 q_\Omega + T_a s_3^2 q_a & \frac{T_a^2}{2}s_2 s_3 q_a & -\frac{T_a^3}{3}\dot{x}\dot{y} q_\Omega + T_a s_3 s_4 q_a & -\frac{T_a^2}{2}\dot{y} q_\Omega \\ \frac{T_a^3}{3}s_1 s_2 q_a & \frac{T_a^2}{2}s_2 s_3 q_a & \frac{T_a^3}{3}s_2^2 q_a & \frac{T_a^2}{2}s_2 s_4 q_a & 0 \\ \frac{T_a^2}{2}s_1 s_4 q_a & -\frac{T_a^3}{3}\dot{x}\dot{y} q_\Omega + T_a s_3 s_4 q_a & \frac{T_a^2}{2}s_2 s_4 q_a & \frac{T_a^3}{3}\dot{x}^2 q_\Omega + T_a s_4^2 q_a & \frac{T_a^2}{2}\dot{x} q_\Omega \\ 0 & -\frac{T_a^2}{2}\dot{y} q_\Omega & 0 & \frac{T_a^2}{2}\dot{x} q_\Omega & T_a q_\Omega \end{bmatrix}. \quad (70)$$

where $\mathbf{v}_c^i(k) \sim \mathcal{N}(0, I_4)$ and

$$D_a^i[\mathbf{x}_a^i(k)] = S_a^i[\mathbf{x}_a^i(k)]V_a^i(k) \quad (61)$$

with

$$S_a^i[\mathbf{x}_a^i(k)] = \begin{bmatrix} \sqrt{q_a}s_1(k) & 0 & 0 & 0 \\ 0 & \sqrt{q_a}s_3(k) & -\sqrt{q_\Omega}\dot{y}(k) & 0 \\ \sqrt{q_\Omega}s_2(k) & 0 & 0 & 0 \\ 0 & \sqrt{q_a}s_4(k) & \sqrt{q_\Omega}\dot{x}(k) & 0 \\ 0 & 0 & 0 & \sqrt{q_\Omega} \end{bmatrix} \quad (62)$$

$$V_a^i(k) = \begin{bmatrix} \sqrt{T_a^3/3} & 0 & 0 & 0 \\ \frac{\sqrt{3T_a}}{2} & \frac{\sqrt{T_a}}{2} & 0 & 0 \\ 0 & 0 & \sqrt{T_a^3/3} & 0 \\ 0 & 0 & \frac{\sqrt{3T_a}}{2} & \frac{\sqrt{T_a}}{2} \end{bmatrix} \quad (63)$$

Guideline for Choice of Target State-Dependent Process Noise Intensity for the NCT Model

Observing the diagonal elements of $Q_a^i[\mathbf{x}_a^i(k)]$ in (71), the process noise induced RMS change rate in the velocity (X- and Y-coordinate) and in the turn rate over a sampling interval T_a are

$$d_a^x \triangleq \frac{\sqrt{\frac{T_a^3}{3}\dot{y}^2 q_\Omega + T_a s_3^2 q_a}}{T_a} \quad (71)$$

$$d_a^y \triangleq \frac{\sqrt{\frac{T_a^3}{3}\dot{x}^2 q_\Omega + T_a s_4^2 q_a}}{T_a} \quad (72)$$

$$d_\Omega \triangleq \frac{\sqrt{T_a q_\Omega}}{T_a} \quad (73)$$

respectively. It is obvious that the choices of the design values d_a (d_a^x and d_a^y) and d_Ω need to be considered simultaneously.

To make things simpler, we consider an extreme case with $\dot{x} = 0$ (similar case for $\dot{y} = 0$). Given a target with speed V_s (which is equal to the magnitude of $\dot{y}$ in this

extreme case), we have

$$s_1^* = 0 \tag{74}$$

$$s_2^* = 1 \tag{75}$$

$$s_3^* = -T_a\Omega(k) \tag{76}$$

$$s_4^* = 1. \tag{77}$$

This leads to

$$d_a^{x*} = \frac{\sqrt{\frac{T_a^3}{3}V_s^2 q_\Omega + T_a^3\Omega^2(k)q_a}}{T_a} \tag{78}$$

$$d_a^{y*} = \frac{\sqrt{T_a q_a}}{T_a}. \tag{79}$$

Since the assumption is $\dot{x} = 0$, the process noise induced RMS change in X-coordinate velocity should be more sensitive compared with that in Y-coordinate. Given a conjectured choice as d_Ω and d_a, the process noise induced RMS change in X-coordinate velocity contributed by the PSD q_Ω and q_a are,

$$d_a^{x*}(q_\Omega, q_a = 0) = \frac{\sqrt{\frac{T_a^3}{3}V_s^2 q_\Omega}}{T_a} = \frac{T_a\dot{y}d_\Omega^*}{\sqrt{3}} \tag{80}$$

$$d_a^{x*}(q_\Omega = 0, q_a) = \frac{\sqrt{T_a^3\Omega^2(k)q_a}}{T_a} = \sqrt{T_a\Omega^2(k)q_a}. \tag{81}$$

It is important to compare those two values. The process noise induced RMS change rate for both the linear velocity and the turn rate should be within reasonable ranges over time interval T_a.

REMARKS

Without loss generality, based on (80), (81) and (82), the selections of the design values d_a and d_Ω for the NCT model are shown as (sequentially)

i) Select d_Ω, then $d_a^{x*}(q_\Omega, q_a = 0)$ has been obtained and should be in a reasonable range.

ii) Select d_a, then $d_a^{x*}(q_\Omega = 0, q_a)$ has been obtained and should be in a reasonable range. Further, d_a^{x*} should be not too big and d_a^{y*} should be not too small.

APPENDIX C. THE MC RESULTS FOR THE SAMPLE CROSSCORRELATION

The *sample crosscorrelation coefficient* between the lth component of $\mathbf{x}^i$ and the hth component of $\mathbf{x}^j$ in M MC runs at a particular point in time (not indicated, for conciseness) is

$$\hat{\rho}^M_{\mathbf{x}_l^i\mathbf{x}_h^j} \triangleq \frac{\sum_{m=1}^M(\hat{\mathbf{x}}_{l,m}^i - \mathbf{x}_l^i)(\hat{\mathbf{x}}_{h,m}^j - \mathbf{x}_h^i)}{\sqrt{\left[\sum_{m=1}^M(\hat{\mathbf{x}}_{l,m}^i - \mathbf{x}_l^i)^2\right]\left[\sum_{m=1}^M(\hat{\mathbf{x}}_{h,m}^j - \mathbf{x}_h^j)^2\right]}}. \tag{82}$$

The sample crosscorrelation coefficients of different heterogeneous components from 1000 MC runs, for the scenario described in Section 5, are shown in Figs. 8–11. It can be seen that the "common process noise effect," driven by real maneuvers here, leads to significant crosscorrelation between the estimation errors from the heterogeneous local sensors. Furthermore, both positive and negative crosscorrelations are observed. This motivates the geometry-based "functional model" discussed in Appendix D.

APPENDIX D. AN APPROXIMATION TECHNIQUE FOR THE CROSSCOVARIANCE MATRIX

By considering the steady-state case for a KF, an approximation technique for the evaluation of the crosscovariance matrix in the homogeneous case has been developed recently [5]. This technique, which relies on the crosscorrelation coefficients between the local estimates of the same state components, namely, position and velocity and the maneuvering indices at the different sensors, can be extended to heterogeneous case as follows.

The components of the state $\mathbf{x}^i$ are grouped by coordinates ($c = 1,2$) as follows

$$\mathbf{x}_1^i = [x \quad \dot{x}]' \tag{83}$$

$$\mathbf{x}_2^i = [y \quad \dot{y}]' \tag{84}$$

which can be "aligned" with those of $\mathbf{x}^j$ as it will be shown in the sequel. Then the first components of $\mathbf{x}_c^i$, $c = 1,2$ and $\mathbf{x}^j$ are position and the second components are velocity, albeit in different spaces. The covariance submatrix P_c^i, corresponding to estimate $\hat{\mathbf{x}}_c^i$, follows from the corresponding elements of P^i.

Extending the crosscovariance matrix approximation technique for the homogeneous case in [5] to the heterogeneous case, we can then approximately reconstruct the crosscovariance matrix elements using the following expression

$$P_{c,lh}^{ij} = a_{c,lh}^{ij}\rho_{c,lh}^{ij}\sqrt{P_c^i(l,l)P^j(h,h)}, \qquad c = 1,2, \quad l,h = 1,2 \tag{85}$$

where $c = 1,2$ correspond to the first and the second part of the state $\mathbf{x}^i$; $l,h = 1$ represent position and $l,h = 2$ represent velocity; $\rho_{c,lh}^{ij}$ is the *maximum crosscorrelation coefficient* of the estimation errors in $\mathbf{x}_{c,l}^i$ and $\mathbf{x}_h^j$ and $a_{c,lh}^{ij}$ is geometry-dependent *adjustment factor* in the crosscorrelation coefficient. This factor is discussed below.

The maximum crosscorrelation coefficients are denoted as ρ_{pp} for position-position, ρ_{pv} for position-velocity, ρ_{vp} for velocity-position and ρ_{vv} for velocity-

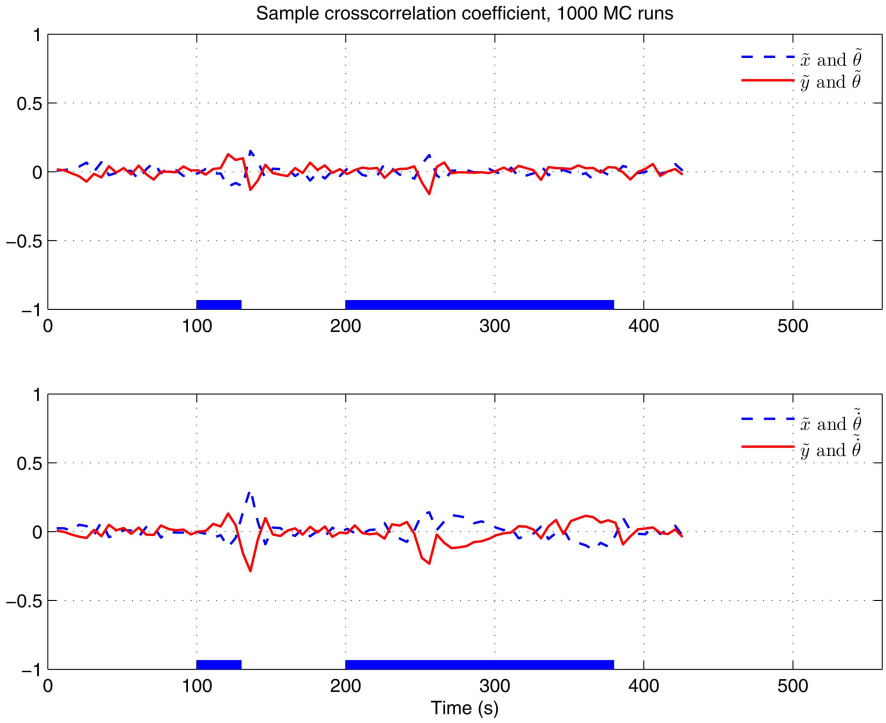


Fig. 8. The sample crosscorrelation for $\tilde{x}$ and $\tilde{y}$ with $\tilde{\theta}$ and $\dot{\tilde{\theta}}$.

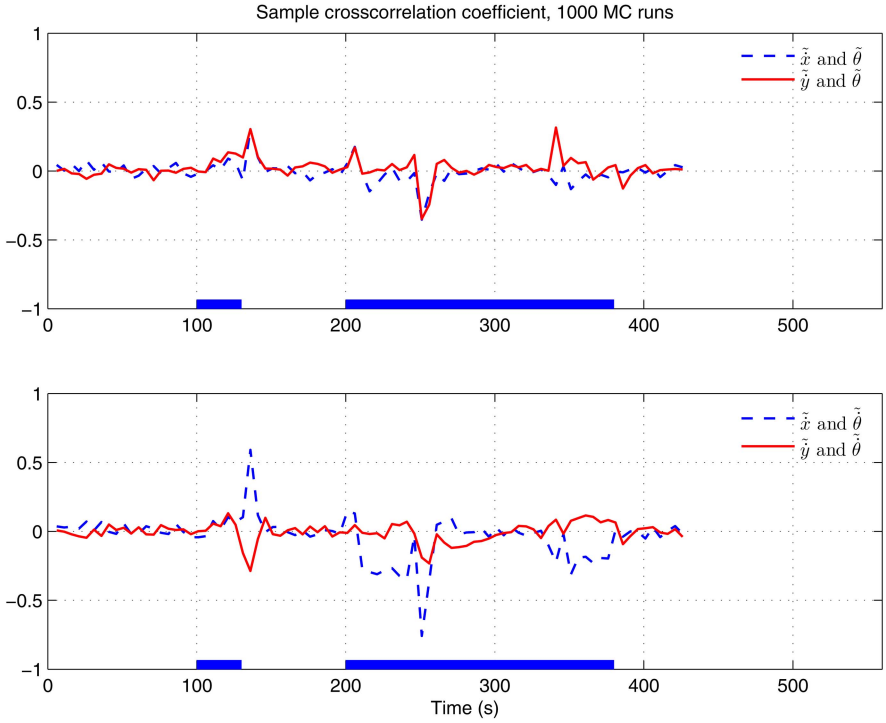


Fig. 9. The sample crosscorrelation for $\dot{\tilde{x}}$ and $\dot{\tilde{y}}$ with $\tilde{\theta}$ and $\dot{\tilde{\theta}}$.

velocity. Then we have

$$\rho_{c,lh}^{ij} = \begin{cases} \rho_{pp} & l,h = 1 \\ \rho_{pv} & l = 1, \quad h = 2 \\ \rho_{vp} & l = 2, \quad h = 1 \\ \rho_{vv} & l,h = 2 \end{cases} \tag{86}$$

which are chosen (similarly to [5]) as $\rho_{pp} = 0.1$, $\rho_{pv} = \rho_{vp} = 0.15$ and $\rho_{vv} = 0.45$ in our simulations.

There are two state variable pairs, say, x-θ and y-θ, for the position-position crosscorrelation coefficient. The best way to quantify the crosscorrelation difference (accounts for the geometry) for those state variable pairs is based on the function (7).

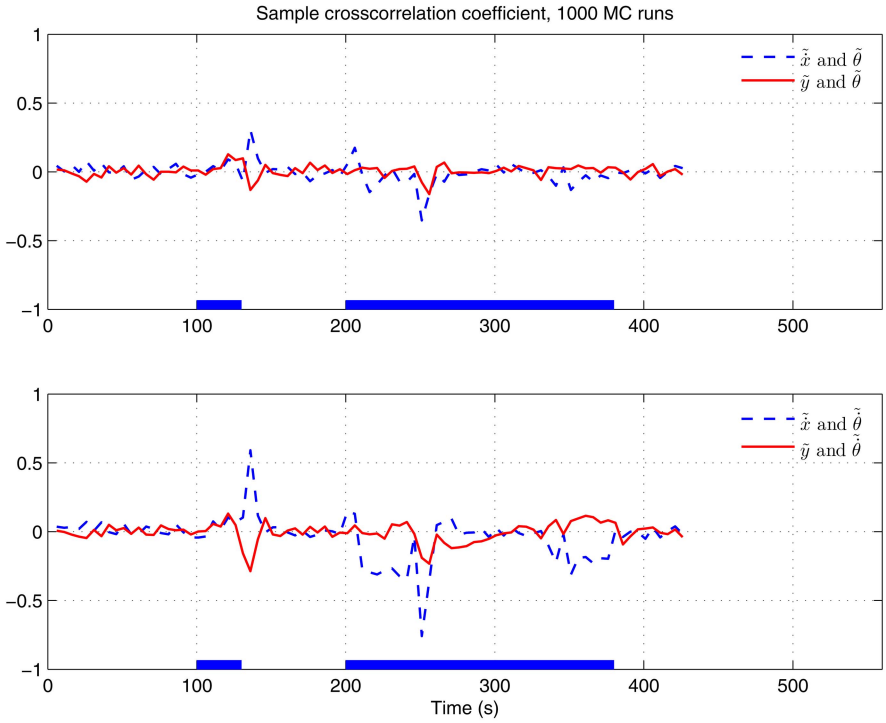


Fig. 10. The sample crosscorrelation for $\tilde{x}$ and $\tilde{y}$ with $\tilde{\theta}$ and $\dot{\tilde{\theta}}$.

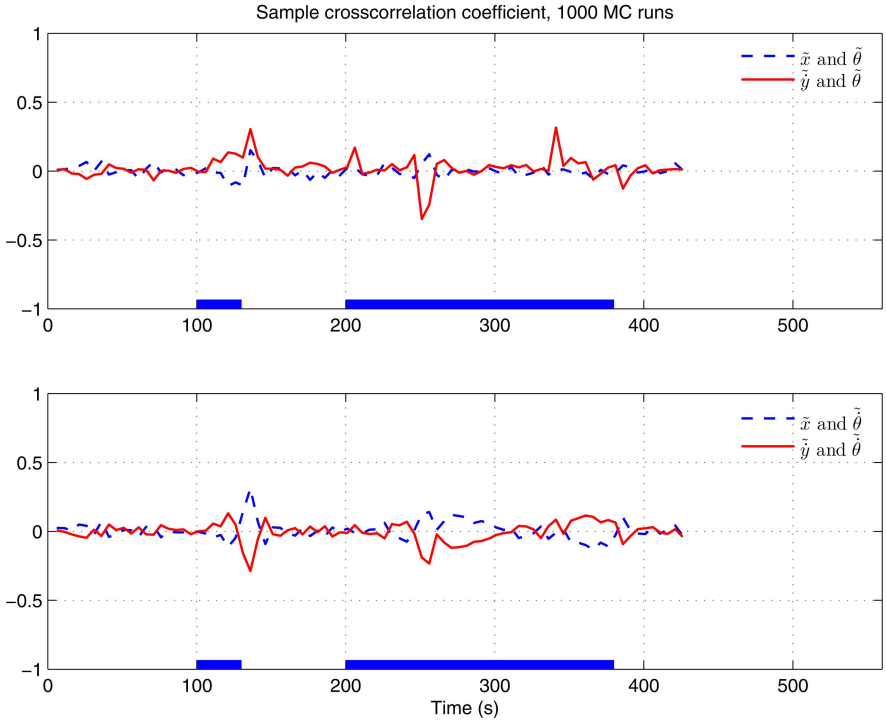


Fig. 11. The sample crosscorrelation for $\tilde{x}$ and $\dot{\tilde{y}}$ with $\tilde{\theta}$ and $\dot{\tilde{\theta}}$.

From the Cartesian to polar coordinate transformation (see, e.g., [4]), one has the following proportionalities

$$\frac{\partial x}{\partial \theta} \propto -\sin\theta \tag{87}$$

$$\frac{\partial y}{\partial \theta} \propto \cos\theta. \tag{88}$$

Based on the above, we have the following adjustment factors for the crosscorrelation coefficients of the pairs x-θ and y-θ

$$a^{ij}_{1,11} = -\sin\theta \tag{89}$$

$$a^{ij}_{2,11} = \cos\theta. \tag{90}$$

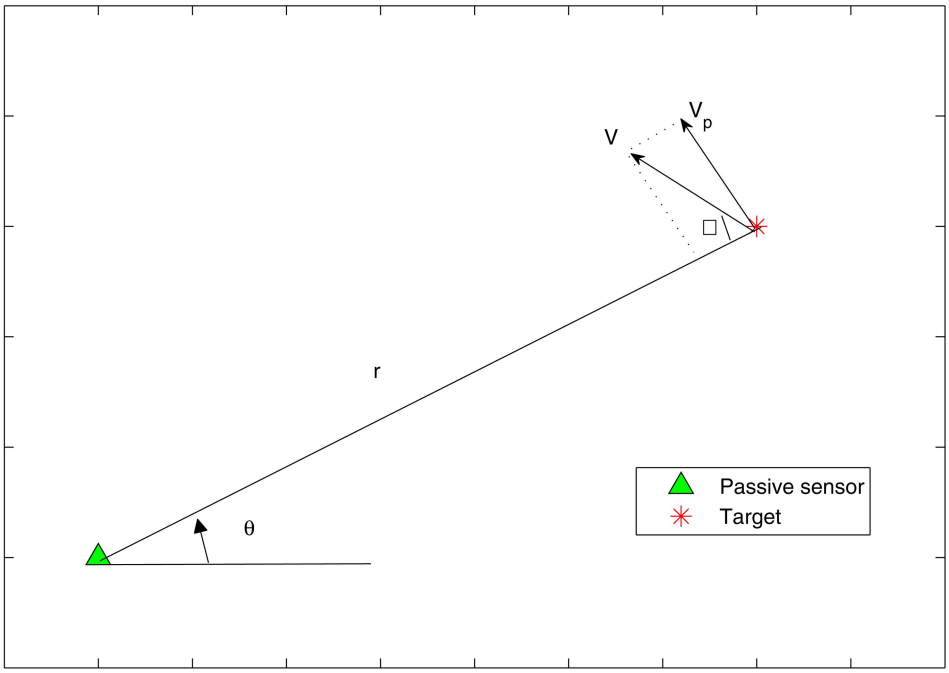


Fig. 12. The illustrated scenario for calculating target maneuvering index.

TABLE II
The RMSE in Position for LMMSE Fuser

	100 s	110 s	130 s	150 s	255 s
"uncorr"	27.6 m	37.5 m	33.6 m	28.9 m	30.3 m
"corr"	26.6 m	36.4 m	33.5 m	29.1 m	29.5 m

TABLE III
The RMSE in Velocity for LMMSE Fuser

	100 s	110 s	130 s	150 s	255 s
"uncorr"	9.4 m/s	15.3 m/s	10.7 m/s	5.1 m/s	17.4 m/s
"corr"	9.5 m/s	14.9 m/s	10.8 m/s	6.1 m/s	18.2 m/s

Then the crosscorrelation coefficients of the pairs x-θ and y-θ are obtained from combining the adjustment factor (90)–(91) with (87) as

$$\rho_{x\theta} = -\rho_{pp}\sin\theta \tag{91}$$

$$\rho_{y\theta} = \rho_{pp}\cos\theta. \tag{92}$$

Similarly, we have the adjustment factors

$$a_{c,lh}^{ij} = \begin{cases} -\sin(\theta), & c = 1 \\ \cos(\theta), & c = 2 \end{cases} \tag{93}$$

and the other crosscorrelation coefficients are

$$\rho_{x\dot{\theta}} = -\rho_{pv}\sin\theta \tag{94}$$

$$\rho_{y\dot{\theta}} = \rho_{pv}\cos\theta \tag{95}$$

$$\rho_{\dot{x}\theta} = -\rho_{vp}\sin\theta \tag{96}$$

$$\rho_{\dot{y}\theta} = \rho_{vp}\cos\theta \tag{97}$$

$$\rho_{\dot{x}\dot{\theta}} = -\rho_{vv}\sin\theta \tag{98}$$

$$\rho_{\dot{y}\dot{\theta}} = \rho_{vv}\cos\theta. \tag{99}$$

Finally, the reconstructed crosscovariance matrix is

$$P^{ij} = \begin{bmatrix} P_1^{ij} \\ P_2^{ij} \end{bmatrix} \tag{100}$$

with the elements of each block in (101) given by (86) with (88)–(100).

For the scenario described in Section 5, the RMSE in position and velocity under both "uncorr" and "corr" assumptions at some fusion times for the LMMSE fuser are shown in Table II and III, respectively.

Based on these results, which show negligible performance differences, it seems preferable to follow the "uncorr" assumption.

APPENDIX E. THE CHOICE OF ESTIMATOR FOR THE PASSIVE SENSOR

The guideline for deciding whether to use an IMM estimator or a (single model) KF can be quantified in terms of the target maneuvering index, which is the ratio between the standard deviation (RMS values) of the motion uncertainty and the measurement uncertainty [2][7]. Namely, if this index is below 0.5 then there is no point in using an IMM.

For the passive sensor considered, the maneuvering index can be calculated as follows. As shown in Fig. 12, the angular velocity seen by the passive sensor is

$$\dot{\theta}_p = \frac{V\sin\varphi}{r_p} \tag{101}$$

where V is the speed of the target and r_p is the range of the target from the passive sensor. Then the angular

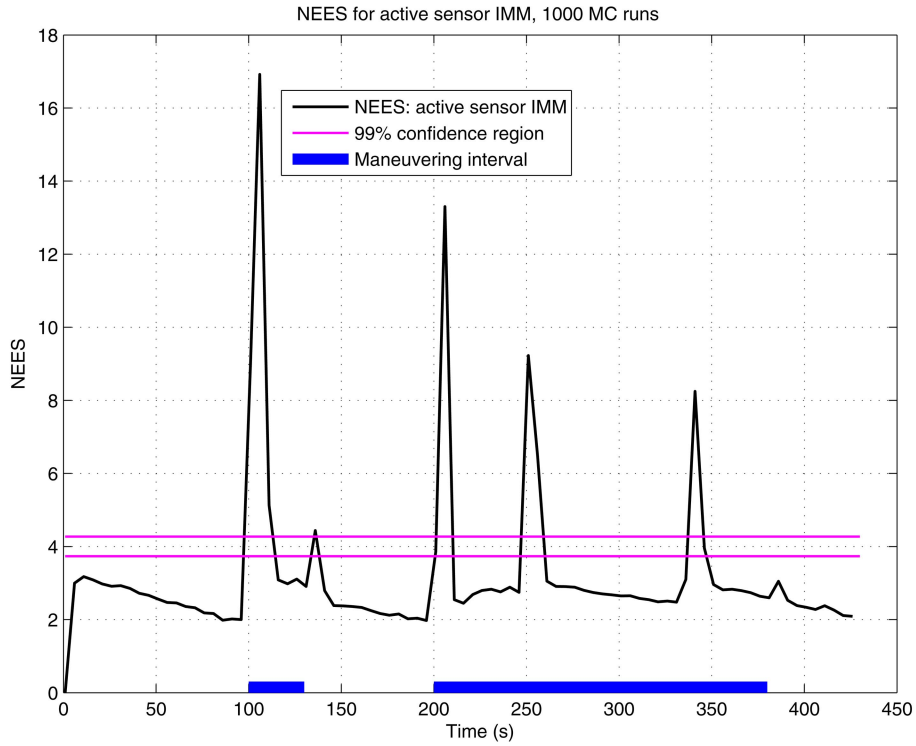


Fig. 13. The NEES for active sensor IMM.

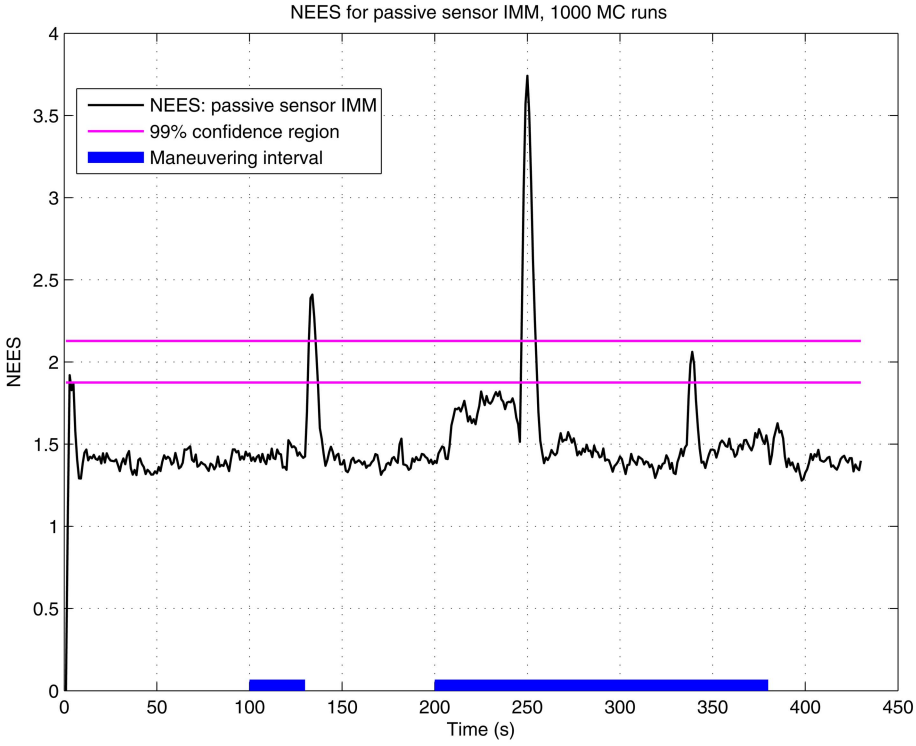


Fig. 14. The NEES for passive sensor KF.

acceleration seen by the passive sensor is

$$\ddot{\theta}_p = \frac{V\cos\varphi}{r_p}\dot{\varphi} \tag{102}$$

where $\dot{\varphi}$ is the target turn rate.

The RMS effect of (103) on the (angular) position, i.e., the angular displacement over sampling interval T_p (multiplied by 2) is $\ddot{\theta}_p T_p^2$. The (target's true) maneuvering index, with the passive sensor noise SD σ_p (in radians), is the (physically dimensionless)

quantity

$$\lambda_p = \frac{\ddot{\theta}_p T_p^2}{\sigma_p} = \frac{\dot{\varphi} T_p^2 V \cos\varphi}{\sigma_p r_p}. \quad (103)$$

For the scenario described in the simulation section, with $V = 250$ m/s, $\cos\varphi \approx 0.8$, $r_p \approx 5 \cdot 10^4$ m, $T_p = 1$ s, $\sigma_p = 1$ mrad and $\dot{\varphi} \approx 30$ mrad/s (which is the maximum target turn rate in our simulation scenario), we have $\lambda_p \approx 0.12$. This small target maneuvering index (less than 0.5) leads to the choice of a KF for the passive sensor, as done in Section 5.

APPENDIX F. THE LOCAL ESTIMATORS' CONSISTENCY

The NEES for the active sensor's IMM and for the passive sensor's KF are shown in Figs. 13 and 14 (the results are obtained by using the estimates and the corresponding covariance matrices for the heterogeneous T2TF, that is, four components' information from the active sensor and two components' information from the passive sensor), respectively. The lack of consistency of the passive sensor KF is due to the maneuvers. The lack of consistency of the active sensor IMM is common and this is due to its (unavoidable delay) in the adaptation. The IMM estimator is "pessimistic" during the no-maneuver intervals and "optimistic" when a maneuver starts or ends until it "catches up." This is the typical behavior of the IMM, which is still superior to any single-model based filter.

REFERENCES

[1] Y. Bar-Shalom and H. Chen
Covariance reconstruction for track fusion with legacy track sources.
Journal of Advances in Information Fusion, **3**, 2 (Dec. 2008), 107–117.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction.
Wiley, 2001.

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

[4] K. C. Chang, R. K. Saha, and Y. Bar-Shalom
On optimal track-to-track fusion.
IEEE Transactions on Aerospace and Electronic Systems, **33**, 4 (Oct. 1997), 1271–1276.

[5] H. Chen, T. Kirubarajan, and Y. Bar-Shalom
Performance limits of track-to-track fusion versus centralized estimation: Theory and application.
IEEE Transactions on Aerospace and Electronic Systems, **AES-39**, 2 (Apr. 2003), 386–398.

[6] H. Chen and Y. Bar-Shalom
Track association and fusion with heterogeneous local trackers.
Proceedings of 46th IEEE Conference on Decision and Control, New Orleans, LA, Dec. 2007.

[7] T. Kirubarajan and Y. Bar-Shalom
Kalman filter versus IMM estimator: When do we need the latter?
IEEE Transactions on Aerospace and Electronic Systems, **39**, 4 (Oct. 2003), 1452–1457.

[8] M. R. Morelande and N. J. Gordon
Target tracking through a coordinated turn.
Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05), vol. 4, iv/21–iv/24, 2005.

[9] R. K. Saha
Track-to-track fusion with dissimilar sensors.
IEEE Transactions on Aerospace and Electronic Systems, **32**, 3 (July 1996), 1021–1029.

[10] X. Tian and Y. Bar-Shalom
Track-to-track fusion configurations and association in a sliding window.
Journal of Advances in Information Fusion, **4**, 2 (Dec. 2009), 146–164.

[11] X. Tian and Y. Bar-Shalom
The optimal algorithm for asynchronous track-to-track fusion.
Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, #7698-46, Orlando, FL, Apr. 2010.

[12] T. Yuan, Y. Bar-Shalom, P. K. Willett, and D. Hardiman
Impact point prediction for short range thrusting projectiles.
Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, #7698-55, Orlando, FL, Apr. 2010.

[13] T. Yuan, Y. Bar-Shalom, P. K. Willett, et. al
A multiple IMM approach with unbiased mixing for thrusting projectiles.
Submitted to *IEEE Transactions on Aerospace and Electronic Systems*.

Ting Yuan "TinYe" received B.S. and M.S. degrees both from the University of Electronic Science and Technology of China. He is currently pursuing the Ph.D. degree in electrical and computer engineering, University of Connecticut, Storrs.

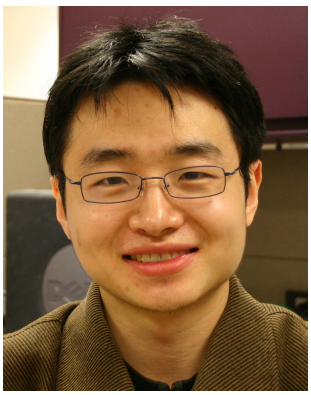
His research interests include statistical signal processing and multisensor track/information fusion.

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

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

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

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



Xin Tian received the B.S. degree in 2002 and M.S. degree in 2005, both from the Department of Information and Communication Engineering, Xi'an Jiaotong University, China. In 2010 he received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Connecticut, Storrs.

Dr. Tian's research areas include statistical signal processing, tracking and information fusion algorithms, detection theory, decision theory under uncertainty, sensor management. He is currently a research scientist at I-fusion Inc., Germantown, MD.

Characterization and Empirical Evaluation of Bayesian and Credal Combination Operators

ALEXANDER KARLSSON
RONNIE JOHANSSON
STEN F. ANDLER

We address the problem of combining independent evidences from multiple sources by utilizing the Bayesian and credal combination operators. We present measures for degree of conflict and imprecision, which we use in order to characterize the behavior of the operators through a number of examples. We introduce discounting operators that can be used whenever information about the reliability of sources is available. The credal discounting operator discounts a credal set with respect to an interval of reliability weights, hence, we allow for expressing reliability of sources imprecisely. We prove that the credal discounting operator can be computed by using the extreme points of its operands. We also perform two experiments containing different levels of risk where we compare the performance of the Bayesian and credal combination operators by using a simple score function that measures the informativeness of a reported decision set. We show that the Bayesian combination operator performed on centroids of operand credal sets outperforms the credal combination operator when no risk is involved in the decision problem. We also show that if a risk component is present in the decision problem, a simple cautious decision policy for the Bayesian combination operator can be constructed that outperforms the corresponding credal decision policy.

Manuscript received August 8, 2010; revised April 27, 2011 and August 17, 2011; released for publication August 22, 2011.

Refereeing of this contribution was handled by Huimin Chen.

Authors addresses: A. Karlsson, Informatics Research Centre, University of Skövde, Sweden, E-mail (alexander.karlsson@his.se); R. Johansson, Informatics Research Centre, University of Skövde, Sweden, E-mail (ronnie.johansson@his.se); S. F. Andler, Informatics Research Centre, University of Skövde, Sweden, E-mail (sten.f.andler@his.se).

1557-6418/11/$17.00

1. INTRODUCTION

Bayesian theory [5] is one of the most commonly utilized theories for managing uncertainty in *information fusion* [20, 12]. The theory relies on two main assumptions: (1) a *probability function* should be used for representing *belief* and (2) *Bayes' theorem* should be used for *belief updating* when a new observation has been made. The main criticism of Bayesian theory that can be found in the literature (e.g., [14, 25]) is that the first assumption is unrealistically strong since one is forced to quantify belief precisely even if one only possesses scarce information about the environment of interest. For this reason, a family of alternative theories has been introduced that usually goes under the name *imprecise probability* [26], where belief can be expressed *imprecisely*.

One common theory that belongs to the family of imprecise probability is *credal set theory* [2, 3, 9, 10, 19], also known as "theory of credal sets" [11] and "quasi-Bayesian theory" [8], where one utilizes a *closed convex set of probability functions* (instead of a single function), denoted as a *credal set* [19], for representing belief. An attractive feature of credal set theory is that it reduces to Bayesian theory if singleton sets are adopted. Furthermore, credal set theory can be thought of as point-wise application of Bayes theorem on all probability (and likelihood) functions within operand sets (unlike, e.g., evidence theory [23], which is inconsistent with this point-wise Bayesian paradigm [2, 3]). Hence, credal set theory can be seen as the most straightforward generalization of Bayesian theory to imprecise probability.

In this paper, we are interested in contrasting Bayesian theory with credal set theory when used for *combining independent pieces of evidence*, known as the *combination problem* [16]. Arnborg [2, 3] has previously characterized the relation between *robust Bayesian theory*, which can be seen as a *sensitivity interpretation* [4, 14] of credal set theory, and *evidence theory* [23] when used for the combination problem. We extend Arnborg's work by characterizing the *Bayesian and credal combination operators*[1] in terms of *imprecision* and *conflict* and by introducing methods for accounting for *reliability* of sources. In addition, we also empirically evaluate the use of the operators for decision making regarding some state space of interest. Since the credal combination operator is considerably more computational demanding than the Bayesian counterpart, such a evaluation can reveal whether or not the additional computational expense yields an increase in decision performance.

[1] Arnborg [2, 3] denoted this operator by "robust Bayesian combination operator." We deliberately avoid using this terminology since robust Bayesianism imposes a sensitivity interpretation of the credal set [4, 14] and we do not want to exclude other interpretations (see e.g., Walley [25]).

The paper is organized in the following way: In Section 2, we derive the Bayesian and credal combination operators. In Section 3, we present measures for conflict and imprecision for the operators. Based on these measures, we present a number of examples that highlight the behavior of the operators. We introduce discounting operators for the Bayesian and credal combination operators, which can be used whenever information about reliability of sources is known. In Section 4, we present two experiments; one where no risk component is present in the decision problem, i.e., there is no cost for making an erroneous decision, and one where such a component exists. We discuss the design and analyze the result of each experiment. Lastly, in Section 5, we summarize the article and present the main conclusions.

2. PRELIMINARIES

We derive the Bayesian and credal combination operators and elaborate on how the credal combination operator can be computed.

2.1. Bayesian Combination Operator

Let X and $Y_1,\dots,Y_n$ be discrete random variables with state spaces Ω_X and $\Omega_{Y_1},\dots,\Omega_{Y_n}$, respectively. Assume that we have n sources and that source $i \in \{1,\dots,n\}$ has made observation $y_i \in \Omega_{Y_i}$ and reported a *likelihood function* $p(y_i \mid X)$ as a representation of the *evidence* provided by y_i regarding X. By assuming that the observations are *conditionally independent given* X, we can construct the *joint evidence* (or joint likelihood):

$$p(y_1,\dots,y_n \mid X) = p(y_1 \mid X)\dots p(y_n \mid X). \tag{1}$$

In principle, we can use (1) as a Bayesian way of *combining the evidences*, however, this is not convenient when implemented in an operational system since the joint evidence monotonically decreases with the number of sources n. Let us therefore elaborate on how this problem can be solved. Let

$$p_i(X) \triangleq \frac{p(y_i \mid X)}{\sum_{x\in\Omega_X} p(y_i \mid x)} \tag{2}$$

i.e., $p_i(X)$ are probability functions (normalized likelihood functions). By using Bayes' theorem and the assumption of conditional independence, we obtain

$$\begin{aligned} p(X \mid y_1,\dots,y_n) &= \frac{p(y_1,\dots,y_n \mid X)p(X)}{\sum_{x\in\Omega_X} p(y_1,\dots,y_n \mid x)p(x)} \\ &= \frac{p(y_1 \mid X)\dots p(y_n \mid X)p(X)}{\sum_{x\in\Omega_X} p(y_1 \mid x)\dots p(y_n \mid x)p(x)} \\ &= \frac{p_1(X)\dots p_n(X)p(X)}{\sum_{x\in\Omega_X} p_1(x)\dots p_n(x)p(x)} \\ &= \frac{\frac{p_1(X)\dots p_n(X)}{\sum_{x\in\Omega_X} p_1(x)\dots p_n(x)}p(X)}{\sum_{x\in\Omega_X}\frac{p_1(x)\dots p_n(x)}{\sum_{x\in\Omega_X} p_1(x)\dots p_n(x)}p(x)}. \end{aligned} \tag{3}$$

Let

$$\Phi(p_1(X),\dots,p_n(X)) \triangleq \frac{p_1(X)\dots p_n(X)}{\sum_{x\in\Omega_X} p_1(x)\dots p_n(x)}. \tag{4}$$

From (3) we see that the joint evidence $p(y_1,\dots,y_n \mid X)$ has the same effect on the *posterior* $p(X \mid y_1,\dots,y_n)$, irrespective of the prior $p(X)$, as $\Phi(p_1(X),\dots,p_n(X))$, i.e., $p(y_1,\dots,y_n \mid X)$ and $\Phi(p_1(X),\dots,p_n(X))$ are equivalent evidences. The following theorem allows us to *recursively combine* evidences into a joint evidence.

THEOREM 1

$$\begin{aligned} &\Phi(\dots\Phi(p_1(X),p_2(X))\dots,p_n(X)) \\ &\quad = \frac{p_1(X)\dots p_n(X)}{\sum_{x\in\Omega_X} p_1(x)\dots p_n(x)}. \end{aligned} \tag{5}$$

PROOF See Appendix.

Note that the normalization in each combination in the recursion eliminates the problem of a monotonically decreasing joint evidence when n increases. We use the recursive form of Φ as our basis for the definition of a *Bayesian combination operator* denoted by $\Phi_{\mathcal{B}}$ (i.e., we define the operator for two operands) [3, 2]:

DEFINITION 1 The Bayesian combination operator is defined as

$$\Phi_{\mathcal{B}}(p_1(X),p_2(X)) \triangleq \frac{p_1(X)p_2(X)}{\sum_{x\in\Omega_X} p_1(x)p_2(x)} \tag{6}$$

where $p_i(X)$, $i \in \{1,2\}$, are conditionally independent evidences in the form of probability functions (normalized likelihood functions). The operator is undefined when $\sum_{x\in\Omega_X} p_1(x)p_2(x) = 0$.

Note that the operator is associative and commutative.

2.2. Credal Combination Operator

The credal combination operator, also known as the *robust Bayesian combination operator* (see Footnote 1) [2, 3], can be derived by using *credal set theory* [19, 9, 10, 2, 3]. As we mentioned in the introduction, in credal set theory one represents belief by a *closed convex set of probability functions*. However, one is also allowed to express *evidence* regarding some random variable imprecisely, i.e., instead of a *single likelihood function* as a representation of evidence, as in the Bayesian case, one can adopt a closed convex set of such functions. Combination of such evidences then amounts to applying the Bayesian combination operator point-wise on all possible combinations of functions from the sets. In order to enforce convexity of the posterior result one applies the *convex-hull operator*.

One important concept within credal set theory, which we will use extensively in the proofs is *convex combination* defined as [1].

DEFINITION 2 A convex combination of probability functions $p_1(X),\ldots,p_n(X)$ is a probability function expressed in the following form

$$\lambda_1 p_1(X) + \cdots + \lambda_n p_n(X) \tag{7}$$

where $(\forall i \in \{1,\ldots,n\})$ $(0 \leq \lambda_i)$ and $\sum_{i=1}^{n} \lambda_i = 1$.

We can now define the *convex hull* of a finite set as the set of all convex combinations of points in the set [1].

DEFINITION 3 The convex hull of a finite set $\{p_1(X),\ldots, p_n(X)\}$ is defined as

$$\begin{aligned}\mathcal{CH}(\{p_1(X),\ldots,p_n(X)\}) &\triangleq \left\{ \sum_{i=1}^{n} \lambda_i p_i(X) : (\forall i \in \{1,\ldots,n\}) \right. \\ &\quad \left. \times (\lambda_i \geq 0), \sum_{i=1}^{n} \lambda_i = 1 \right\}. \end{aligned} \tag{8}$$

Let $\mathcal{P}(X)$ denote a *prior credal set*, i.e., a closed convex set of probability functions of the form $p(X)$ and $\mathcal{P}(X \mid y)$ a *posterior credal set* of functions $p(X \mid y)$. Let $\mathcal{E}(\mathcal{P}(X))$ denote the set of *extreme points* of $\mathcal{P}(X)$, i.e., points that belong to the set and cannot be expressed as a convex combination of other points in the set. We are now ready to define the notion of independence for credal sets referred to as *strong independence* [7].

DEFINITION 4 The discrete random variables X and Y are strongly independent iff all $p(X,Y) \in \mathcal{E}(\mathcal{P}(X,Y))$ can be expressed as $p(X,Y) = p(X)p(Y)$, where $p(X) \in \mathcal{P}(X)$ and $p(Y) \in \mathcal{P}(Y)$. Similarly, X and Y are strongly conditionally independent given Z iff all $p(X,Y \mid z) \in \mathcal{E}(\mathcal{P}(X,Y \mid z))$ can be expressed as $p(X,Y \mid z) = p(X \mid z) \cdot p(Y \mid z)$, $\forall z \in \Omega_Z$, where $p(X \mid z) \in \mathcal{P}(X \mid z)$ and $p(Y \mid z) \in \mathcal{P}(Y \mid z)$.

The intuition behind this definition is that each extreme point of a joint credal set should fulfill the same criteria for independence as in ordinary probability calculus, i.e., the extreme points should factorize [11]. By using this notion of independence, the credal combination operator[2] can be derived as a straightforward generalization of the Bayesian combination operator.

DEFINITION 5 The credal combination operator is defined as

$$\begin{aligned}\Phi_{\mathcal{C}}(\mathcal{P}_1(X),\mathcal{P}_2(X)) &\triangleq \mathcal{CH}(\{\Phi_{\mathcal{B}}(p_1(X),p_2(X)) : p_1(X) \in \mathcal{P}_1(X), \\ &\quad p_2(X) \in \mathcal{P}_2(X)\}) \end{aligned} \tag{9}$$

where $\mathcal{P}_i(X)$, $i \in \{1,2\}$, are strongly conditionally independent evidences in the form of credal sets (closed convex sets of normalized likelihood functions) and where $\mathcal{CH}$ is the convex-hull operator. The $\Phi_{\mathcal{C}}$ operator is undefined iff there exists $p_i(X) \in \mathcal{P}_i(X)$, $i \in \{1,2\}$, such that $\Phi_{\mathcal{B}}$ is undefined.

The operator is associative and commutative. Note that the operator is based on point-wise application of the Bayesian combination operator on all combinations of functions from the operand credal sets. Hence, the operator is equivalent to the Bayesian combination operator for singleton sets. One important credal set, that we will use extensively throughout the article, is the set of all probability functions for a given state space, denoted as a *probability simplex*.

DEFINITION 6 The probability simplex $\mathcal{P}^*(X)$ for a discrete random variable X with state space Ω_X is defined as

$$\mathcal{P}^*(X) \triangleq \left\{ p(X) : (\forall x \in \Omega_X)(p(x) \geq 0), \sum_{x \in \Omega_X} p(x) = 1 \right\} \tag{10}$$

In order to compute the credal combination operator, we only consider operand credal sets that has a finite number of extreme points. Such a property can be guaranteed by using credal sets in the form of *polytopes* [1]

DEFINITION 7 A credal set $\mathcal{P}(X)$ is a polytope iff

$$\mathcal{P}(X) = \mathcal{CH}(\{p_1(X),\ldots,p_n(X)\}) \tag{11}$$

where $\{p_1(X),\ldots,p_n(X)\} \subset \mathcal{P}^*(X)$ is a finite set and where $\mathcal{CH}$ is the convex-hull operator.

The following theorem enables computation by extreme points of the credal combination operator when the operands are polytopes (the theorem was implicitly mentioned by Arnborg [3], with no proof, and explicitly stated by Arnborg [2], but only a "proof hint" was provided. A corresponding theorem has been stated and proved for filtering (continuous case) by Noack, et al. [21, Theorem 2]).

THEOREM 2

$$\begin{aligned}\Phi_{\mathcal{C}}(\mathcal{P}_1(X),\mathcal{P}_2(X)) &= \Phi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}_1(X)),\mathcal{E}(\mathcal{P}_2(X))). \end{aligned} \tag{12}$$

PROOF See Appendix.

3. CHARACTERIZATION OF THE BAYESIAN AND CREDAL COMBINATION OPERATORS

In this section,[3] we define measures for *degree of conflict* and *imprecision* and use these for characterizing the behavior the Bayesian and credal combination operators through a number of examples. We introduce

[2]Arnborg [2, 3] defined the operator without the inclusion of a convex-hull operator, however, he mentions in the discussion following his definition that such operator should be utilized. See also Footnote 1.

[3]This section includes material from Karlsson et al. [16].

discounting operators that can be used whenever information about the *reliability* of sources is known. We exemplify the utilization of the discounting operators by revisiting the examples.

3.1. Degree of Conflict

One important concept when combining evidences from multiple sources is *the degree of conflict* measured on the evidences reported by the sources. Intuitively, such measure can be thought of as an "inverse similarity measure," i.e., the more similar the reported evidences are, the less conflict exists between the sources. Hence, for the Bayesian case, we simply use the *Euclidean norm* as the basis for a conflict measure.

DEFINITION 8 The degree of conflict between two evidences in the form of probability functions $p_1(X)$ and $p_2(X)$ is defined as

$$\Gamma_{\mathcal{B}}(p_1(X), p_2(X)) \triangleq \frac{\|p_1(X) - p_2(X)\|}{\sqrt{2}} \tag{13}$$

where $\|\cdot\|$ is the Euclidean norm and where the denominator constitutes the diameter of the probability simplex $\mathcal{P}^*(X)$, i.e.,

$$\max\left\{\max_{p_j(X)\in\mathcal{P}^*(X)} \|p_i(X) - p_j(X)\| : p_i(X) \in \mathcal{P}^*(X)\right\} = \sqrt{2}. \tag{14}$$

Similarly to the above Bayesian conflict measure, we base a credal conflict measure on the notion of similarity. A similarity measure for general closed convex sets exists under the name of *Hausdorff distance* [15]. The Hausdorff distance is the largest distance one can find between a point from any of the two sets to the closest point in the other set. By using the Hausdorff distance we can define the following conflict measure for credal sets:

DEFINITION 9 The degree of conflict between two credal sets $\mathcal{P}_1(X)$ and $\mathcal{P}_2(X)$ is defined as

$$\Gamma_{\mathcal{C}}(\mathcal{P}_1(X), \mathcal{P}_2(X)) \triangleq \frac{\mathcal{H}(\mathcal{P}_1(X), \mathcal{P}_2(X))}{\sqrt{2}} \tag{15}$$

where the denominator constitutes the diameter of the probability simplex $\mathcal{P}^*(X)$ and where $\mathcal{H}$ is the Hausdorff distance defined by [15]

$$\mathcal{H}(\mathcal{P}_1(X), \mathcal{P}_2(X)) \triangleq \max\{\vec{\mathcal{H}}(\mathcal{P}_1(X), \mathcal{P}_2(X)), \vec{\mathcal{H}}(\mathcal{P}_2(X), \mathcal{P}_1(X))\} \tag{16}$$

where $\vec{\mathcal{H}}$ is the forward Hausdorff distance:

$$\vec{\mathcal{H}}(\mathcal{P}_1(X), \mathcal{P}_2(X)) \triangleq \max\left\{\min_{p_2(X)\in\mathcal{P}_2(X)} \|p_1(X) - p_2(X)\| : p_1(X) \in \mathcal{P}_1(X)\right\} \tag{17}$$

where $\|\cdot\|$ is the Euclidean norm.

Note that the credal conflict measure reduces to the Bayesian conflict measure for singleton sets. The forward Hausdorff-distance can be calculated in $O(|\mathcal{E}(\mathcal{P}_1(X))||\mathcal{F}(\mathcal{P}_2(X))|)$ [15], where $\mathcal{F}(\mathcal{P}(X))$ is the set of faces of $\mathcal{P}(X)$.

3.2. Degree of Imprecision

Obviously, since credal set theory belongs to the family of theories referred to as *imprecise probabilities* [26], *imprecision* is an important concept to define. Walley [25, Section 5.1.4] has introduced a measure which he refers to as *the degree of imprecision* for an event $x \in \Omega_X$

$$\Delta(x, \mathcal{P}(X)) \triangleq \max_{p(X)\in\mathcal{P}(X)} p(x) - \min_{p(X)\in\mathcal{P}(X)} p(x). \tag{18}$$

However, the measure does not capture the imprecision of a credal set since it only operates on single events. Let us therefore base our measure of *degree of imprecision* for a credal set on a simple average of Walley's measure in the following way.

DEFINITION 10 The degree of imprecision of a credal set $\mathcal{P}(X)$ is defined as

$$\mathcal{I}(\mathcal{P}(X)) \triangleq \frac{1}{|\Omega_X|} \sum_{x\in\Omega_X} \Delta(x, \mathcal{P}(X)) \tag{19}$$

where $\Delta(x, \mathcal{P}(X))$ is Walley's measure for degree of imprecision for a single event [25, Section 5.1.4].

As an example, if we have a credal set $\mathcal{P}(X)$ where

$$\mathcal{P}(X) = \mathcal{CH}(\{(0.2, 0.1, 0.7)^T, (0.5, 0.2, 0.3)^T, (0.6, 0.3, 0.1)^T\}) \tag{20}$$

where $\Omega_X = \{x_1, x_2, x_3\}$ and the order of the probabilities is $(p(x_1), p(x_2), p(x_3))^T$, then

$$\begin{aligned}\mathcal{I}(\mathcal{P}(X)) &= \frac{1}{3}\sum_{i=1}^{3} \Delta(x_i, \mathcal{P}(X)) \\ &= \frac{1}{3}(0.6 - 0.2 + 0.3 - 0.1 + 0.7 - 0.1) \\ &= 0.4.\end{aligned} \tag{21}$$

3.3. Examples

We here provide a number of examples, containing different degrees of imprecision and conflict, in order to characterize the behavior of the Bayesian and credal combination operators. Let us first elaborate on a convenient way of visualizing belief and evidence in cases where the state space consists of three elements. Assume that $\Omega_X = \{x_1, x_2, x_3\}$. In such case the probability simplex $\mathcal{P}^*(X)$ constitutes the plane orthogonally projected on two dimensional space, seen in Fig. 1, which is geometrically equivalent to the convex hull of the points $(1,0,0)^T$, $(0,1,0)^T$, and $(0,0,1)^T$. Each corner of the triangle represents an extreme point of $\mathcal{P}^*(X)$, i.e., a probability function where all probability mass lies on

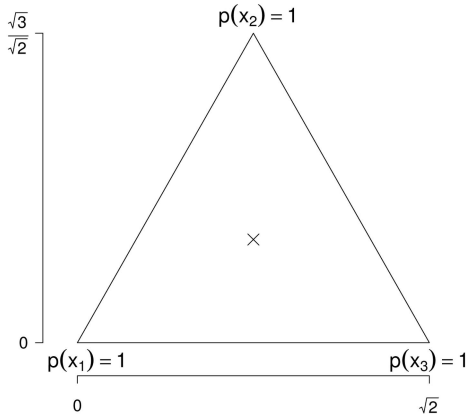


Fig. 1. Probability simplex $\mathcal{P}^*(X)$, where $\Omega_X = \{x_1, x_2, x_3\}$, projected on two dimensional space.

a single element of Ω_X. Each point in the triangle represents a probability function. As an example, the center of the triangle, indicated with a cross, is the uniform distribution over Ω_X. The closer a specific point is to one of the corners in the triangle the higher probability for the respective state. This type of visualization is commonly used within the imprecise probability community (see, e.g., [25]).

3.3.1. Bayesian Combination Operator

Let us start with the example seen in Fig. 2 where there only is a minor conflict among the sources. We see that since both sources suggest x_2 as most probable, seen in Fig. 2(a), the joint evidence, seen in Fig. 2(b), is reinforced towards this state. Now, consider the case where there is a strong conflict among the sources instead, seen in Fig. 3. Both sources have provided evidences that states that x_3 is unlikely to be the true state of X. However, there is a strong disagreement, i.e., conflict among the sources, regarding the states x_1 and x_2. The joint evidence, seen in Fig. 3(b), is approximately uniformly distributed over $\{x_1, x_2\}$, i.e., from the result we cannot single out a best choice between these two states, however, x_3 is still highly unlikely due to the distance to that corner.

3.3.2. Credal Combination Operator

Let us again start with an example where there is a low degree of conflict between the sources, seen in Fig. 4. The operand credal sets in Fig. 4(a) have been constructed by expanding equilateral triangles around the operands in Fig. 2(a). From the figure we see that both sources essentially agree on the state x_2 as being most probable. Therefore the combined evidence

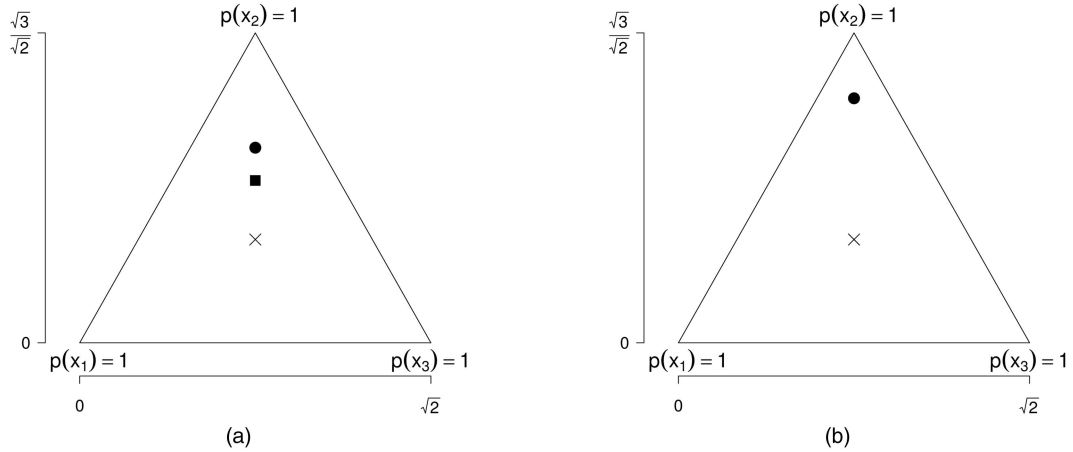


Fig. 2. $p_1(X)$, $p_2(X)$, and $p_{1:2}(X)$ when a low degree of conflict is present. (a) $p_1(X)$ (circle) and $p_2(X)$ (square). (b) $p_{1:2}(X)$.

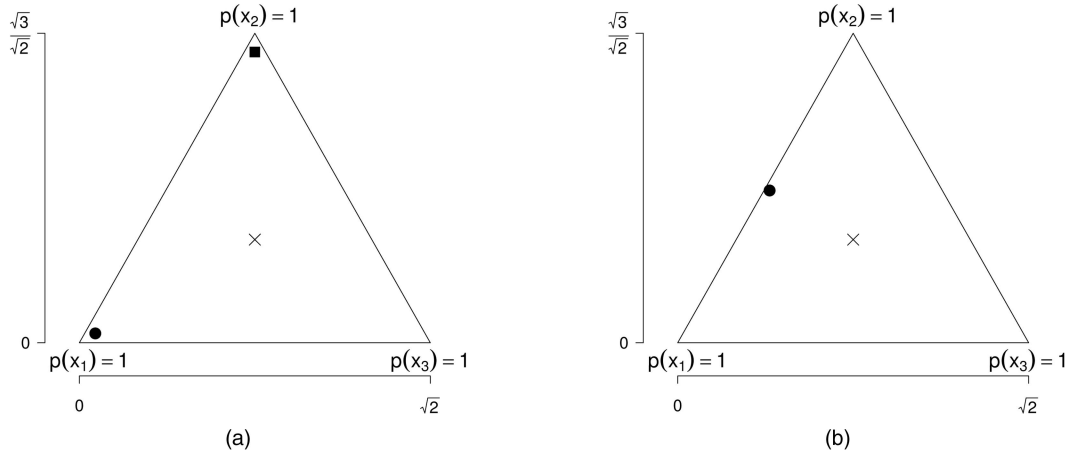


Fig. 3. $p_1(X)$, $p_2(X)$, and $p_{1:2}(X)$ when a high degree of conflict is present. (a) $p_1(X)$ (circle) and $p_2(X)$ (square). (b) $p_{1:2}(X)$.

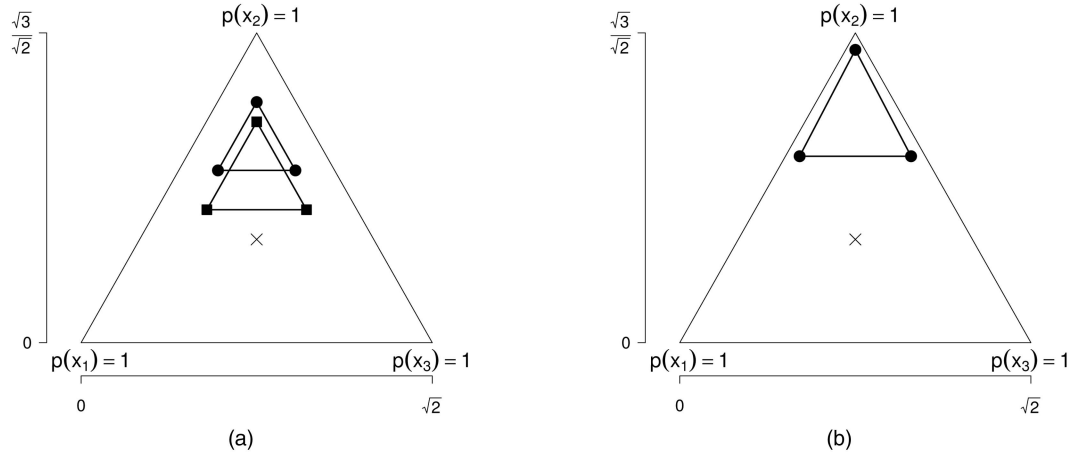


Fig. 4. $\mathcal{P}_1(X)$, $\mathcal{P}_2(X)$, and $\mathcal{P}_{1:2}(X)$ when a low degree of conflict is present. (a) $\mathcal{P}_1(X)$ (circles) and $\mathcal{P}_2(X)$ (squares). (b) $\mathcal{P}_{1:2}(X)$.

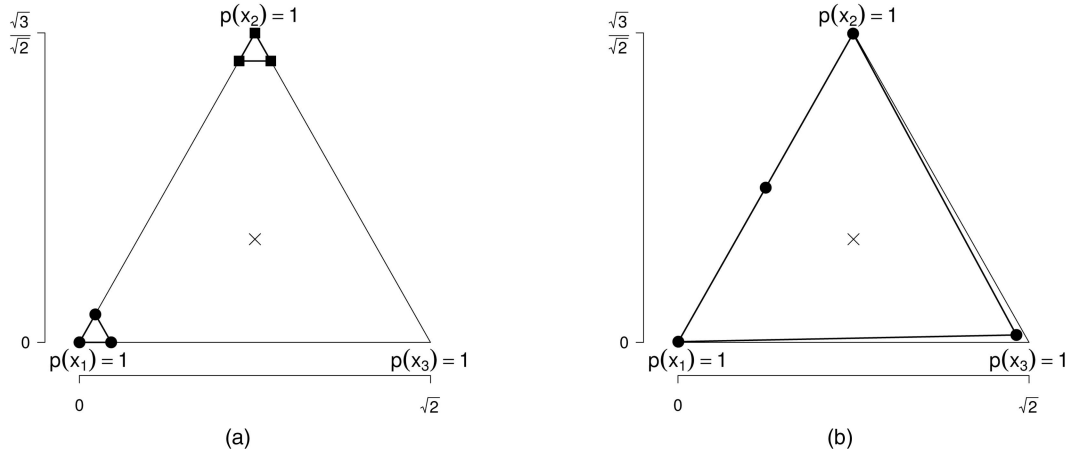


Fig. 5. $\mathcal{P}_1(X)$, $\mathcal{P}_2(X)$, and $\mathcal{P}_{1:2}(X)$ when a high degree of conflict is present. (a) $\mathcal{P}_1(X)$ (circles) and $\mathcal{P}_2(X)$ (squares). (b) $\mathcal{P}_{1:2}(X)$.

$\mathcal{P}_{1:2}(X)$ is reinforced towards a high probability for the state x_2, as is seen in Fig. 4(b). Note that $\mathcal{P}_{1:2}(X)$ preserves the property of not favoring any of the states x_1 and x_3.

Consider again an example where evidences are strongly conflicting (a similar example has been presented by Arnborg [2]). The evidences provided by the sources can be seen in Fig. 5. We see that the resulting joint evidence has a high degree of imprecision. Note that it is the combination of the lower right extreme points of the operand credal sets that is the cause for the lower right extreme point of the joint evidence; a case that has similarities with the well-known *Zadeh's example* for *Dempster's combination rule* [27]. This is due to that the extreme points componentwise suppress each other for the states x_1 and x_2, i.e., if we denote the lower right extreme point of $\mathcal{P}_1(X)$, $\mathcal{P}_2(X)$, and $\mathcal{P}_{1:2}(X)$ by $p_1(X)$, $p_2(X)$, and $p_{1:2}(X)$, respectively, where

$$p_1(x_1) = 1 - \epsilon - \vartheta, \qquad p_1(x_2) = \epsilon, \qquad p_1(x_3) = \vartheta \tag{22}$$

$$p_2(x_1) = \epsilon, \qquad p_2(x_2) = 1 - \epsilon - \vartheta, \qquad p_2(x_3) = \vartheta \tag{23}$$

then we obtain the following expression for $p_{1:2}(x_3)$

$$p_{1:2}(x_3) = \frac{\vartheta^2}{(1-\epsilon-\vartheta)\epsilon + \epsilon(1-\epsilon-\vartheta) + \vartheta^2} \tag{24}$$

which approaches one when $\epsilon \to 0$ (in the figure, $\epsilon > 0$, which is why the lower right extreme point of $\mathcal{P}_{1:2}(X)$ is not exactly positioned at the lower right corner of the probability simplex $\mathcal{P}^*(X)$).

Lastly, let us consider another type of conflict that can appear in the credal case, seen in Fig. 6, where one of the sources expresses a credal set that is highly imprecise, i.e., approximately equivalent to the probability simplex $\mathcal{P}^*(X)$ (there is a small distance between the extreme points of the credal set and the extreme points of the probability simplex which cannot be seen from the figure), and the other source expresses a credal set that constitutes strong evidence for the state x_2. Since the highly imprecise credal set contains probability functions that constitute strong evidence for each of the states in Ω_X, such a credal set is not significantly affected by other operands, unless these contain probability functions that are considerably stronger.[4]

[4]Arnborg [3] denotes the probability simplex as "total scepticism," since such a set is impossible to affect.

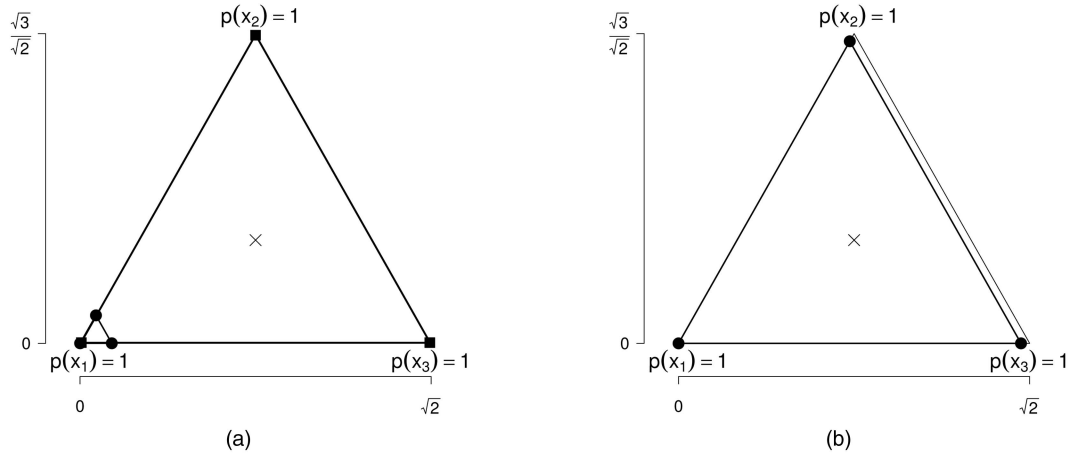


Fig. 6. $\mathcal{P}_1(X)$, $\mathcal{P}_2(X)$ and $\mathcal{P}_{1:2}(X)$ when a high degree of conflict is present. (a) $\mathcal{P}_1(X)$ (circles) and $\mathcal{P}_2(X)$ (squares). (b) $\mathcal{P}_{1:2}(X)$.

3.4. Discounting

In cases where a strong conflict is present among the sources that provide evidences, as was the case for several examples in the previous section, it may be beneficial to account for the sources' *reliability*. If one has obtained information regarding the reliability of sources, e.g., in terms of sensor quality, then it would be reasonable to compensate for such information prior to the combination. Intuitively, if a source is less reliable, then that source should have less effect on the end result, i.e., the joint evidence should be less influenced by that source. Accounting for the reliability of sources is commonly referred to as *discounting* in the literature [23].

3.4.1. Bayesian Discounting Operator

Discounting with respect to the Bayesian combination operator is performed by transforming an operand to a "more uniform" probability function. The reason for this is that the uniform probability function represents an evidence that does not affect the joint evidence in any way:[5]

DEFINITION 11 The Bayesian discounting operator for an evidence in the form of a probability function $p(X)$ with state space Ω_X is defined as

$$\Psi_{\mathcal{B}}(p(X), w) \triangleq wp(X) + (1-w)p_u(X) \tag{25}$$

where $w \in [0,1]$ is a reliability weight, describing a degree of reliability of the discounted source, and $p_u(X)$ is the uniform distribution over Ω_X.

Let us now revisit the example in Fig. 3 but where we have obtained the following reliability weights for the sources[6]

$$w_1 = 0.85, \qquad w_2 = 0.95. \tag{26}$$

Let us introduce the following short-hand notation

$$p_{w_i}(X) \triangleq \Psi_{\mathcal{B}}(p_i(X), w_i) \tag{27}$$

where $i \in \{1,2\}$ and

$$p_{w_1,w_2}(X) \triangleq \Phi_{\mathcal{B}}(\Psi_{\mathcal{B}}(p_1(X), w_1), \Psi_{\mathcal{B}}(p_2(X), w_2)). \tag{28}$$

The result of applying the Bayesian discounting operator with the reliability weights in (26) is seen in Fig. 7. In contrast to the former case, where no discounting was performed, we can here see that due to that the first source is slightly more unreliable, the result is less influenced by that source. This is seen from the figure, since the resulting probability function is closer to the corner $p(x_2) = 1$ than $p(x_1) = 1$.

3.4.2. Credal Discounting Operator

Consider discounting a source that reports an operand credal set for the credal combination operator. Instead of using a single reliability weight, we here allow reliability weights to be expressed imprecisely[7] by a *convex set of reliability weights* $\mathcal{W} \triangleq [\underline{w}, \bar{w}]$ where $[\underline{w}, \bar{w}] \subseteq [0,1]$, i.e., an interval. If we generalize the Bayesian discounting operator to the credal case, we obtain an operator that point-wise discounts each distribution in the credal set with respect to each reliability

[5] Arnborg [2] adopted another interpretation of discounting, which amounts to increase the imprecision for an operand single probability function. We have based our interpretation of discounting on evidence theory [23], i.e., that a discounted operand should have less effect on the end result.

[6] We introduce the Bayesian and credal discounting operators without specifying an exact interpretation of the reliability weights. In principle such a interpretation can differ depending on the application. Exploring different modeling schemas and interpretations for reliabilities is a topic for future research.

[7] Imprecision in reliability weights was inspired by Troffaes [24].

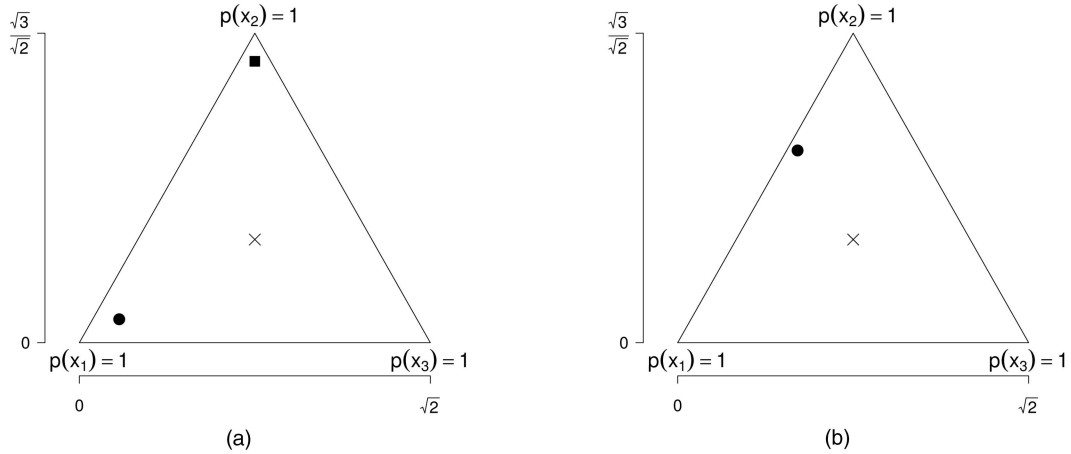


Fig. 7. $p_{w_1}(X)$, $p_{w_2}(X)$, and the discounted combined result $p_{w_1,w_2}(X)$. (a) $p_{w_1}(X)$ (circle) and $p_{w_2}(X)$ (square). (b) $p_{w_1,w_2}(X)$.

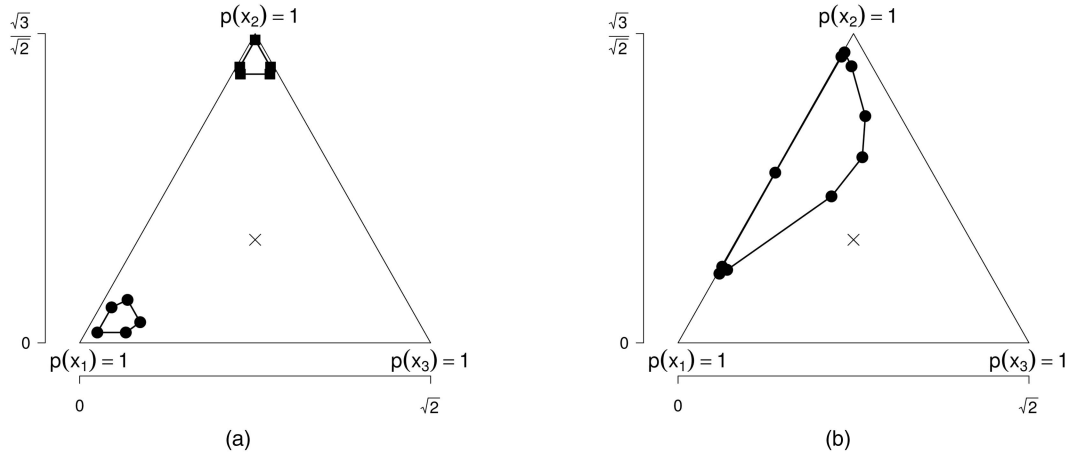


Fig. 8. $\mathcal{P}_{\mathcal{W}_1}(X)$, $\mathcal{P}_{\mathcal{W}_2}(X)$, and $\mathcal{P}_{\mathcal{W}_1,\mathcal{W}_2}(X)$. (a) $\mathcal{P}_{\mathcal{W}_1}(X)$ (circles) and $\mathcal{P}_{\mathcal{W}_2}(X)$ (squares). (b) $\mathcal{P}_{\mathcal{W}_1,\mathcal{W}_2}(X)$.

weight in $\mathcal{W}$:

DEFINITION 12 The discounting operator for a credal set $\mathcal{P}(X)$ given a set of reliability weights $\mathcal{W} = [\underline{w}, \bar{w}]$, where $[\underline{w}, \bar{w}] \subseteq [0,1]$, is defined as

$$\Psi_{\mathcal{C}}(\mathcal{P}(X), \mathcal{W}) \triangleq \mathcal{CH}(\{\Psi_{\mathcal{B}}(p(X), w) : w \in \mathcal{W}, p(X) \in \mathcal{P}(X)\}) \tag{29}$$

where $\Psi_{\mathcal{B}}(p(X), w)$ is the Bayesian discounting operator.

The discounting operator collapses a credal set point-wise towards the uniform distribution. The following theorem allows computation of the credal discounting operator by using the extreme points of the operand sets:

THEOREM 3

$$\Psi_{\mathcal{C}}(\mathcal{P}(X), \mathcal{W}) = \Psi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}(X)), \mathcal{E}(\mathcal{W})). \tag{30}$$

PROOF See Appendix.

Let us now revisit the previous presented examples where a strong conflict was present. Assume that we have obtained the following reliability weights[6] for the example in Fig. 5

$$\mathcal{W}_1 = [0.80, 0.90], \qquad \mathcal{W}_2 = [0.93, 0.98]. \tag{31}$$

Let us introduce some short-hand notation

$$\mathcal{P}_{\mathcal{W}_i}(X) \triangleq \Psi_{\mathcal{C}}(\mathcal{P}_i(X), \mathcal{W}_i) \tag{32}$$

where $i \in \{1,2\}$ and

$$\mathcal{P}_{\mathcal{W}_1,\mathcal{W}_2}(X) \triangleq \Phi_{\mathcal{C}}(\Psi_{\mathcal{C}}(\mathcal{P}_1(X), \mathcal{W}_1), \Psi_{\mathcal{C}}(\mathcal{P}_2(X), \mathcal{W}_2)). \tag{33}$$

The results of applying the discounting operator is seen in Fig. 8. We see that there is a significant difference in terms of imprecision compared to the non-discounted case in Fig. 5(b).

Let us also revisit the example shown in Fig. 6. Assume that one has obtained the following reliabilities for the sources

$$\mathcal{W}_1 = [1.00, 1.00], \qquad \mathcal{W}_2 = [0.75, 0.80]. \tag{34}$$

The result of discounting the sources with respect to these weights is seen in Fig. 9. The lower bound of

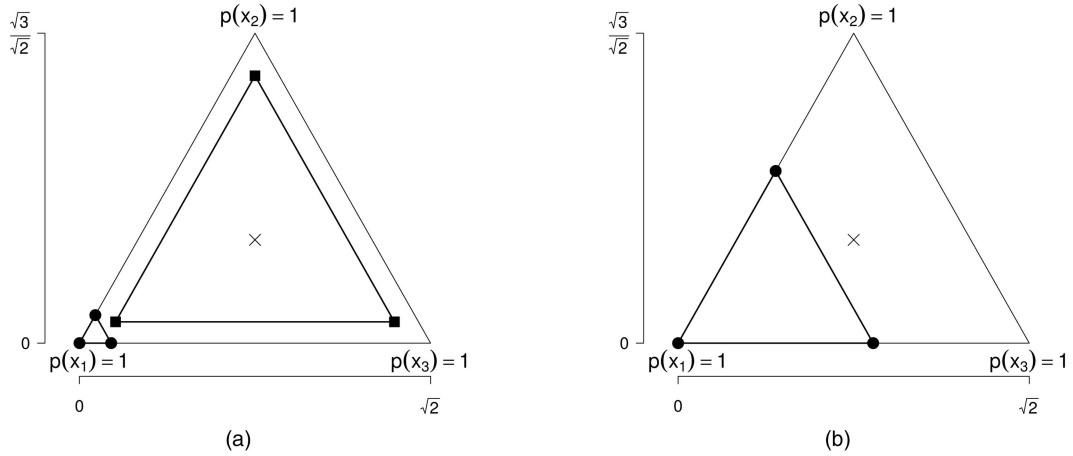


Fig. 9. $\mathcal{P}_{\mathcal{W}_1}(X)$, $\mathcal{P}_{\mathcal{W}_2}(X)$, and $\mathcal{P}_{\mathcal{W}_1,\mathcal{W}_2}(X)$. (a) $\mathcal{P}_{\mathcal{W}_1}(X)$ (circles) and $\mathcal{P}_{\mathcal{W}_2}(X)$ (squares). (b) $\mathcal{P}_{\mathcal{W}_1,\mathcal{W}_2}(X)$.

$\mathcal{W}_2$ will in this case not have any effect since $\mathcal{P}_2(X)$ is centered around the uniform distribution.

4. EMPIRICAL EVALUATION OF BAYESIAN AND CREDAL COMBINATION OPERATORS

In a previous study[8] [17], we explored the performance of the Bayesian and credal combination operators when a single decision is made, i.e., a single state is chosen. In order to select a single decision in the credal case, we selected a representative function from the joint evidence to base the decision upon. We explored three main ways of selecting such a function, often found in the literature [3, 8], namely: a random function, the maximum entropy function, and the centroid function. By assuming that the sources have an implicit uniform second-order distribution over the operand credal set, as a representation of not favoring any probability function within the set, we found that using the Bayesian combination operator on centroids of operand credal sets significantly outperform[9] any of the credal decision schemas (i.e., using the credal combination operator and any of the representative function previously mentioned). The reason for this result is that the second-order distribution can be considerably skewed over the joint evidence and using centroid distributions of operand credal sets as operands for the Bayesian combination operator constitutes a better approximation of the expected value with respect to this skewed second-order distribution over the joint evidence.

Even though it may be better to utilize the Bayesian combination operator than the credal counterpart when a single decision has to be made, the question remains whether or not the credal combination operator can be beneficial to utilize when a set of decisions is allowed. In principle, an optimal method for "set-output" should only output a non-singleton set when the singleton decision output from a Bayesian method is erroneous.

Exploring the performance of utilizing the Bayesian and credal combination operators when decision sets are allowed (i.e., a set of states) is the main aim with our experiments that we will present in the coming sections.[10] In the experiments, we use a simple state space consisting of three states since we then can perform exact computation of the credal combination operator. In real-world applications one is likely to be forced to use some approximations technique in order to limit the number of extreme points of the involved credal sets, since this number can grow exponentially in worst case (with respect to the number of combinations). We present two main experiments for combining evidences reported by a number of sources where there exists some degree of conflict between them. We motivate the utilization of a decision set by a risk component, i.e., there is a large negative cost if one reports a set that does not contain the true state. In the first experiment there is no risk component whereas in the second experiment such a component exists.

4.1. Experiment A—No Risk

Let us start with a scenario that does not contain a risk component in the sense that there is no cost of reporting an erroneous state. Assume that we are interested in determining the state of a random variable X with a state space consisting of three possible states, i.e., $\Omega_X = \{x_1, x_2, x_3\}$, and that we base our decision regarding X on n sources that provide us with pieces of evidence regarding X in the form of strongly conditionally independent credal sets $\mathcal{P}_1(X), \ldots, \mathcal{P}_n(X)$. Assume that the true state of X is x_2. Obviously, if we have selected the sources well, a majority of these provides us with credal sets $\hat{\mathcal{P}}(X)$ that constitute evidence for the

[8] In Karlsson, et al. [17] we considered the problem of belief updating instead of evidence combination, however, it is the same basic operator that is used for both cases.

[9] Two score functions (i.e., performance metrics) were used for comparison: (1) *accuracy* and (2) *Brier loss* [6].

[10] The sections includes material from Karlsson, et al. [18].

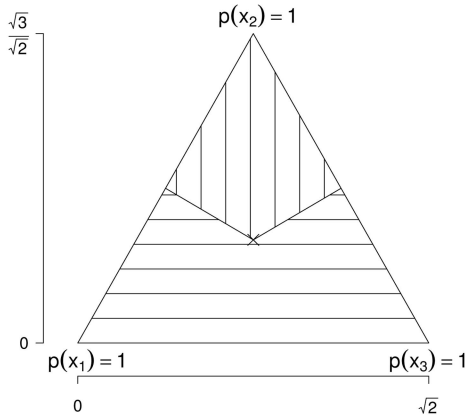


Fig. 10. The probability simplex $\mathcal{P}^*(X)$ partitioned into evidence region (vertical lines) and counter-evidence region (horizontal line) with respect to the true state x_2.

truth solely, i.e.,

$$\hat{p}(X) \in \hat{\mathcal{P}}(X) \Rightarrow x_2 = \arg\max_{x \in \Omega_X} \hat{p}(x). \tag{35}$$

Such type of credal set is completely contained in the region with vertical lines shown in Fig. 10. Let us assume that there is a possibility of obtaining a counter evidence $\tilde{\mathcal{P}}(X)$ with respect to the truth from some of the sources, i.e.,

$$\tilde{p}(X) \in \tilde{\mathcal{P}}(X) \Rightarrow x_2 \neq \arg\max_{x \in \Omega_X} \tilde{p}(x). \tag{36}$$

The counter evidence $\tilde{\mathcal{P}}(X)$ is completely contained in the region with horizontal lines shown in Fig. 10. The imprecision of the credal evidence and counter evidence can be thought of as a *second-order uncertainty* regarding the strength of an evidence in the form of a probability function (i.e., a Bayesian evidence). Let us assume that the sources have no reason to favor any probability function in the credal evidence, i.e., the sources are *indifferent* regarding the probability functions.

Now, assume that we want to combine all the evidences obtained from the sources into a joint evidence. In the Bayesian case, since we cannot apply the $\Phi_{\mathcal{B}}$ operator directly on the operand credal sets, we need to select a single representative probability function from the operands to be utilized for combination. Since the sources are indifferent regarding the probability functions in the operand credal sets, we can assume an implicit uniform distribution over the sets. It is therefore reasonable to utilize the *expected value* of this distribution as a representative function, i.e., *the centroid distribution*. Consequently we obtain the following joint Bayesian evidence

$$p_{1:n}(X) \triangleq \Phi_{\mathcal{B}}(\ldots \Phi_{\mathcal{B}}(\Upsilon(\mathcal{P}_1(X)), \Upsilon(\mathcal{P}_2(X))) \ldots, \Upsilon(\mathcal{P}_n(X))) \tag{37}$$

where the operator Υ is defined as

$$\Upsilon(\mathcal{P}(X)) \triangleq \mathrm{E}_{\mathrm{Un}(\mathcal{P}(X))}[\mathcal{P}(X)] \tag{38}$$

where $\mathrm{Un}(\mathcal{P}(X))$ denotes the uniform distribution over $\mathcal{P}(X)$ (i.e., $\Upsilon(\mathcal{P}(X))$ gives the centroid distribution of $\mathcal{P}(X)$). In the credal case, the joint evidence is straightforwardly obtained by utilizing the $\Phi_{\mathcal{C}}$ operator

$$\mathcal{P}_{1:n}(X) \triangleq \Phi_{\mathcal{C}}(\ldots \Phi_{\mathcal{C}}(\mathcal{P}_1(X), \mathcal{P}_2(X)) \ldots, \mathcal{P}_n(X)). \tag{39}$$

Now, based on the joint Bayesian and credal evidences, we want to make a decision regarding the true state of the variable X. In the Bayesian case this is simply performed by reporting the most probable state(s)

$$\mathcal{D}_{\mathcal{B}}(p_{1:n}(X)) \triangleq \{x_i \in \Omega_X : (\forall x_j \in \Omega_X)(p_{1:n}(x_i) \geq p_{1:n}(x_j))\}. \tag{40}$$

From the above equation, we see that the *Bayesian decision set* $\mathcal{D}_{\mathcal{B}}(p_{1:n}(X))$ is singleton in a majority of the cases. In the credal case, however, it is quite likely, depending on the degree of imprecision reported by the sources, that the decision set is non-singleton

$$\mathcal{D}_{\mathcal{C}}(\mathcal{P}_{1:n}(X)) \triangleq \bigcup_{p_{1:n}(X) \in \mathcal{P}_{1:n}(X)} \mathcal{D}_{\mathcal{B}}(p_{1:n}(X)). \tag{41}$$

Note that unless all probability functions within $\mathcal{P}_{1:n}(X)$ agree on the most probable state, the decision set is non-singleton. Let us also introduce a credal method where the $\Phi_{\mathcal{C}}$ operator is used for constructing the joint credal evidence but where the centroid distribution of the evidence is used for decision making in the same way as in the Bayesian case (see (40)), i.e.,

$$\mathcal{D}^c_{\mathcal{C}}(\mathcal{P}_{1:n}(X)) \triangleq \mathcal{D}_{\mathcal{B}}(\Upsilon(\mathcal{P}_{1:n}(X))). \tag{42}$$

An example of using the $\Phi_{\mathcal{B}}$ and $\Phi_{\mathcal{C}}$ operators is seen in Fig. 11. In Fig. 11(a), we see that one of the sources has reported a quite strong evidence for the truth x_2 while the other source has reported a counter evidence to this state (an evidence for x_1). Fig. 11(b) shows the results of the Bayesian and credal methods. We see that

$$\begin{aligned} \mathcal{D}_{\mathcal{B}}(p_{1:2}(X)) &= \{x_2\} \\ \mathcal{D}_{\mathcal{C}}(\mathcal{P}_{1:2}(X)) &= \{x_1, x_2\} \\ \mathcal{D}^c_{\mathcal{C}}(\mathcal{P}_{1:2}(X)) &= \{x_2\}. \end{aligned} \tag{43}$$

Note that the centroid of the joint credal evidence differs from the joint Bayesian evidence.

Now, obviously a decision set $\mathcal{D} \subseteq \Omega_X$ that contains two states where one of them is the true state, i.e., x_2, should be less valued in comparison to a decision set that is singleton with the true state. Moreover, a decision set that is equal to the state space is clearly non-informative about X since we have already modeled the set of possibilities for X by Ω_X. Hence such decision set is not regarded to be of any value. Based on this

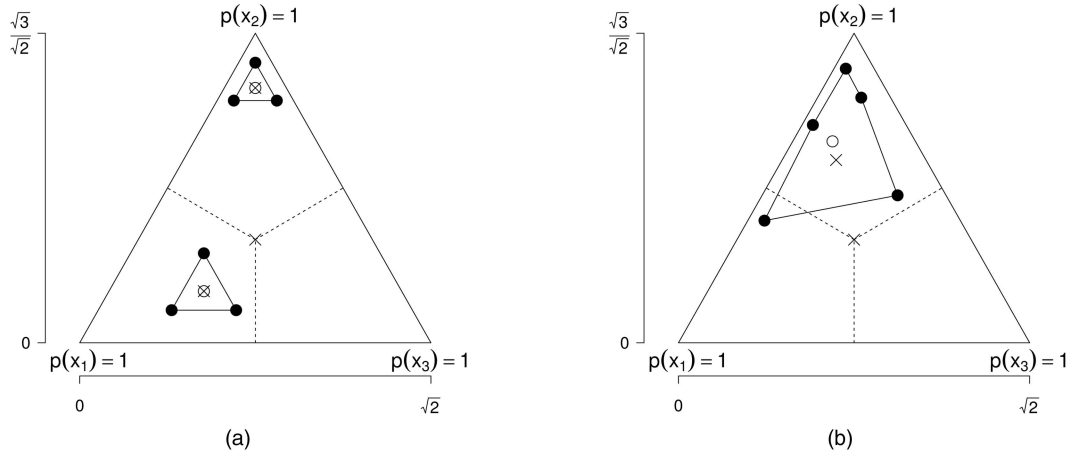


Fig. 11. The example shows the probability simplex $\mathcal{P}^*(X)$ where one operand constitutes evidence for the true state x_2 and the other counter evidence to the true state (in this case evidence for x_1). The dashed lines show the decision regions for the state space. The extreme points of the credal operands $\mathcal{P}_i(X)$, $i \in \{1,2\}$ and joint credal evidence $\mathcal{P}_{1:2}(X)$ are depicted by filled circles. The centroid of $\mathcal{P}^*(X)$ and $\mathcal{P}_{1:2}(X)$ is depicted by a cross. The Bayesian joint evidence $p_{1:2}(X)$ is depicted by an unfilled circle. (a) $\mathcal{P}_i(X)$ and $\mathrm{E}_{\mathrm{Un}(\mathcal{P}_i(X))}[\mathcal{P}_i(X)]$, $i \in \{1,2\}$. (b) $\mathcal{P}_{1:2}(X)$ and $p_{1:2}(X)$.

reasoning we adopt the following *score function* for our experiment

$$f_\alpha(\mathcal{D}) \triangleq \begin{cases} \dfrac{1}{|\mathcal{D}|}, & \text{if} \quad x_2 \in \mathcal{D},\ \mathcal{D} \neq \Omega_X \\ 0, & \text{if} \quad \mathcal{D} = \Omega_X \\ -\alpha, & \text{otherwise} \end{cases} \tag{44}$$

where $\mathcal{D} \subseteq \Omega_X$ and α models a risk component. As we stated in the beginning of this section, we first explore the performance of the methods when no risk is involved in the decision problem, hence, we instantiate the score function with $\alpha = 0$, i.e., $f_0(\mathcal{D})$.

Let the probability for the event that a source reports an evidence with respect to the truth (i.e., x_2) be denoted by β. Note that if we sum the degree of conflict for both the Bayesian and credal conflict measures for all $n-1$ combinations, i.e.,

$$\Gamma_\bullet^{1:n-1} \triangleq \sum_{i=1}^{n-1} \Gamma_\bullet^i \tag{45}$$

where $\Gamma_\bullet^i$ (Definitions 8 and 9) denotes the conflict in the ith combination, then we expected $\Gamma_\bullet^{1:n-1}$ to increase when β monotonically decreases in the interval $[0.5,1]$, i.e., the total amount of conflict among the sources increases. The experiment can now be defined by the following step-wise description:

1. Sample the number of sources $n \sim \mathrm{Un}([5,10])$.
2. Sample the probability for obtaining an evidence for the true state $\beta \sim \mathrm{Un}([0.7,0.9])$.
3. Sample evidences $\mathcal{P}_1(X),\ldots,\mathcal{P}_n(X)$ where the probability of sampling an evidence $\hat{\mathcal{P}}_i(X)$ for the truth is β (see (35)) and $1-\beta$ for a counter evidence $\tilde{\mathcal{P}}_i(X)$ (see (36)), $i \in \{1,\ldots,n\}$.
4. Calculate joint evidences $p_{1:n}(X)$ and $\mathcal{P}_{1:n}(X)$.
5. Calculate decision sets $\mathcal{D}_{\mathcal{B}}(p_{1:n}(X))$, $\mathcal{D}_{\mathcal{C}}(\mathcal{P}_{1:n}(X))$, and $\mathcal{D}_{\mathcal{C}}^{c}(\mathcal{P}_{1:n}(X))$.
6. Calculate the score $f_\alpha(\cdot)$ for each decision set in the previous step.
7. Repeat $m = 10^5$ times.

Remember that we have instantiated $\alpha = 0$ for this first experiment, i.e., there is no risk component involved. Let us elaborate somewhat on the implementation detail of the above description. In step three, we sample evidences by first deciding, utilizing β, if a specific source should report an evidence or a counter evidence for the truth. Then, when we know if it is an evidence or counter evidence that we should sample, we sample a centroid from the corresponding region (see Fig. 10), uniformly. Given the centroid, we sample imprecision by considering the distance from the centroid to the corner points of an equilateral triangle, under the condition that all corner points should reside in the same evidence region. Hence, the credal operand that we sample are all equilateral triangles (simplices) that are completely contained in the evidence or counter-evidence region with respect to the truth (e.g., Fig. 11(a)). Credal sets of this form can be obtained by interval constraints on marginal probabilities.

4.4.1. Results

The results of the experiment is seen in Table I. We see that the expected score of the Bayesian method $\mathcal{D}_{\mathcal{B}}$ is clearly better then the credal method $\mathcal{D}_{\mathcal{C}}$. This means that the credal method does not isolate the cases for which the Bayesian method performs poorly in an optimal way, since we would then have expected a higher score for the credal method than the Bayesian one. This is seen from Table I since in 21.9% of the cases the credal method outputs a non-singleton set while the Bayesian method only outputs an erroneous state in 7.2% of the cases. The credal method outputs a

TABLE I
Expected Score $\mathrm{E}[f_0(\cdot)]$, with 95% Confidence Intervals, for $\mathcal{D}_{\mathcal{B}}(p_{1:n}(X))$, $\mathcal{D}_{\mathcal{C}}(\mathcal{P}_{1:n}(X))$ and $\mathcal{D}^c_{\mathcal{C}}(\mathcal{P}_{1:n}(X))$

Method	$\mathrm{E}[f_0(\cdot)]$	$f_0(\cdot) > 0$ (%)		$f_0(\cdot) = 0$ (%)		
		$\lvert\cdot\rvert = 1$	$\lvert\cdot\rvert = 2$	$\lvert\cdot\rvert = 1$	$\lvert\cdot\rvert = 2$	$\lvert\cdot\rvert = 3$
$\mathcal{D}_{\mathcal{B}}(p_{1:n}(X))$	0.93 ± 0.002	92.8	0.0	7.2	0.0	0.0
$\mathcal{D}_{\mathcal{C}}(\mathcal{P}_{1:n}(X))$	0.85 ± 0.002	78.1	13.1	1.8	0.4	6.6
$\mathcal{D}^c_{\mathcal{C}}(\mathcal{P}_{1:n}(X))$	0.92 ± 0.002	91.6	0.0	8.4	0.0	0.0

decision set of no value (i.e., x_2 is not in the decision set or Ω_X is reported) in 8.8% of the cases. In fact, even if we would let the credal method obtain a reward of one in cases where two states is reported and one of them is the truth, the credal method would still perform worse than the Bayesian method (78.1% + 13.1% = 91.2% compared to 92.8%). Also note that the Bayesian method $\mathcal{D}_{\mathcal{B}}$ performs better than the credal centroid method $\mathcal{D}^c_{\mathcal{C}}$, however, the difference is not as high compared to the former case.

4.2. Experiment B—Risk

One argument that one might have for using the credal method $\mathcal{D}_{\mathcal{C}}$ is that even though it cannot optimally isolate the cases where the Bayesian method $\mathcal{D}_{\mathcal{B}}$ performs poorly, it can still be an interesting choice when there exists a risk component in the decision problem, i.e., reporting an erroneous state is coupled with a negative cost. Indeed, if we use the result from Table I, we see that the Bayesian method reports an erroneous state in 7.2% of the cases while the credal method only makes erroneous reports in 1.8% + 0.4% = 2.2% of the cases. Hence, if we would have set $\alpha = 10$ in the score function in (44), we would have obtained an expected score $\mathrm{E}[f_{10}(\mathcal{D}_{\mathcal{B}}(p_{1:n}(X)))] \approx 0.21$ for the Bayesian method and $\mathrm{E}[f_{10}(\mathcal{D}_{\mathcal{C}}(\mathcal{P}_{1:n}(X)))] \approx 0.63$ for the credal correspondence. However, when risk is incorporated in the decision problem, there clearly exist cases when using the Bayesian method for which one would not simply output the single state that maximizes the probability, e.g., whenever the joint Bayesian evidence $p_{1:n}(X)$ is close to the uniform distribution. Let us therefore modify $\mathcal{D}_{\mathcal{B}}$ to a *cautious Bayesian method* $\mathcal{D}^{\delta}_{\mathcal{B}}$ in the following way

$$\mathcal{D}^{\delta}_{\mathcal{B}}(p_{1:n}(X)) \triangleq \{x \in \Omega_X : p_{1:n}(x) > \delta\} \tag{46}$$

where $\delta \in [0, |\Omega_X|^{-1}]$. The method partitions the probability simplex into decision regions, seen in Fig. 12. Note that a high value of δ yields a less cautious method and vice versa. Note that when $\delta = 0$ we get $\mathcal{D}^0_{\mathcal{B}}(p_{1:n}(X)) = \Omega_X$ for all joint evidences $p_{1:n}(X)$ that do not reside on the boundary of the probability simplex. Also note that when $\delta = |\Omega_X|^{-1}$ we still have decision regions that are non-singleton.

Now let us use the same simulation settings as in Experiment A (Section 4.1) but where we now introduce a risk component by setting $\alpha = 10$, yielding a score function f_{10}. We perform the simulation for at set of values of the parameter $\delta \in [0, |\Omega_X|^{-1}]$ to see if there exist parameters that cause the cautious Bayesian method to outperform the credal method.

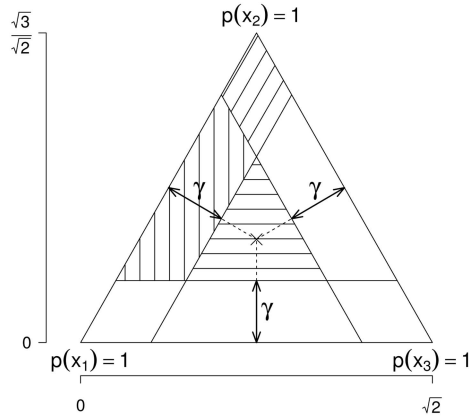


Fig. 12. An example of the cautious Bayesian method in (46) where $\delta = 0.2$. The parameter δ imposes decision regions by planes that are parallel to the (proper) faces of the simplex with a distance $\gamma = \delta(\sqrt{3}/\sqrt{2})$. The horizontal lines depict the decision region for Ω_X, the vertical lines depict $\{x_1, x_2\}$, and lastly the region with diagonal lines depicts $\{x_2\}$.

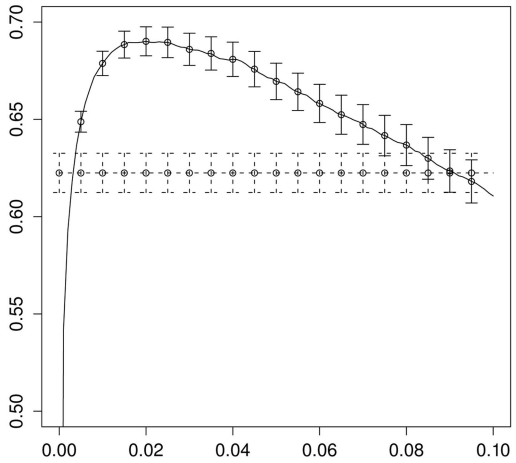


Fig. 13. The solid line shows the cautious Bayesian method $\mathcal{D}^{\delta}_{\mathcal{B}}(p_{1:n}(X))$ and the dashed line the credal method $\mathcal{D}_{\mathcal{C}}(\mathcal{P}_{1:n}(X))$. The x-axis depicts δ and the y-axis $\mathrm{E}[f_{10}(\cdot)]$. Confidence intervals on the 95%-level are also shown.

4.2.1. Results

The result is shown in Fig. 13. The cautious Bayesian method outperforms the credal method when $\delta \in [0.005,$

TABLE II
Expected Score $\mathrm{E}[f_{10}(\cdot)]$, with 95% Confidence Intervals, for the Methods $\mathcal{D}_{\mathcal{B}}^{0.02}(p_{1:n}(X))$ and $\mathcal{D}_{\mathcal{C}}(\mathcal{P}_{1:n}(X))$

Method	$\mathrm{E}[f_{10}(\cdot)]$	$f_{10}(\cdot) \geq 0$ (%)			$f_{10}(\cdot) < 0$ (%)	
		$\lvert\cdot\rvert = 1$	$\lvert\cdot\rvert = 2$	$\lvert\cdot\rvert = 3$	$\lvert\cdot\rvert = 1$	$\lvert\cdot\rvert = 2$
$\mathcal{D}_{\mathcal{B}}^{0.02}(p_{1:n}(X))$	0.69 ± 0.006	69.6	22.1	7.1	0.7	0.5
$\mathcal{D}_{\mathcal{C}}(\mathcal{P}_{1:n}(X))$	0.62 ± 0.010	77.9	13.2	6.7	1.8	0.4

TABLE III
The table shows the cautious Bayesian parameter δ in $\mathcal{D}_{\mathcal{B}}^{\delta}(p_{1:n}(X))$ for different risks α. The intervals for δ depicts the region for which $\mathcal{D}_{\mathcal{B}}^{\delta}(p_{1:n}(X))$ outperforms $\mathcal{D}_{\mathcal{C}}(\mathcal{P}_{1:n}(X))$ with respect to $\mathrm{E}[f_{\alpha}(\cdot)]$ and where the 95% confidence intervals are non-overlapping for the methods.

α	0	2	4	6	8	10
δ	[0.05, 0.33]	[0.04, 0.33]	[0.03, 0.14]	[0.02, 0.10]	[0.01, 0.08]	[0.01, 0.07]

0.07]. Let us explore the cautious Bayesian method at its peak performance, which approximately occur at $\delta = 0.02$. The result of this parameter value is seen in Table II. We see that the cautious Bayesian method tends to output a non-singleton set more often than the credal counterpart. However, the Bayesian method only reports an erroneous state in $0.7\% + 0.5\% = 1.2\%$ of the cases compared to $1.8\% + 0.4\% = 2.2\%$ in the credal case and due to the high risk component this yields a better score for the Bayesian method. Note that since δ is quite low and the cautious Bayesian method only outputs Ω_X in approximately 7.1% of the cases, we can conclude that the joint Bayesian evidence $p_{1:n}(X)$ is close to the boundary of the probability simplex in a majority of the cases, which is quite natural since we have assumed that a majority ($\beta \in [0.7, 0.9]$) of the sources output evidence for the true state x_2. Let us further study the sensitivity of the parameter δ by exploring how a parameter set for δ changes with respect to the risk for the case where the cautious Bayesian method outperforms the credal method (where the 95% confidence intervals are non-overlapping), seen in Table III. From the table we see that when there is a low risk, $\alpha \in \{0, 2\}$, the parameter sets for δ where $\mathcal{D}_{\mathcal{B}}^{\delta}$ outperforms $\mathcal{D}_{\mathcal{C}}$ are quite large. When the risk increases, $\alpha \in \{4, 6, 8, 10\}$, the parameter sets becomes considerably smaller. From the table we see that $\mathcal{D}_{\mathcal{B}}^{\delta}$ with $\delta = 0.05$ performs better than $\mathcal{D}_{\mathcal{C}}$ irrespective of the risk α.

5. SUMMARY AND CONCLUSIONS

We have characterized the behavior of the Bayesian and credal combination operators through a number of examples. We introduced measures for degree of conflict and imprecision and explored the behavior of the Bayesian and credal combination operators for a number of examples where different degrees of conflict and imprecision were present. We highlighted that when a strong conflict is present between the sources that report credal sets, the joint evidence can be highly imprecise. We therefore introduced Bayesian and credal discounting operators that can be utilized whenever information about the reliability of sources is available. We showed that the credal discounting operator can be computed by utilizing the extreme points of the operands (credal set and interval of reliability weights). Finally, we showed that the credal discounting operator can have a significant impact on the combined result when used.

Both the Bayesian and credal discounting operators have been introduced so that they are consistent with the underlying paradigm in the respective theories, i.e., the Bayesian discounting operator takes a single reliability weight as an argument while the credal such operator takes a convex set of reliability weights in the form of an interval. Hence the Bayesian discounting operator assumes that a precise reliability weight can always be formulated while in the credal case imprecision is allowed. Moreover, the credal discounting operator preserves the intuitive paradigm of being a point-wise version of the Bayesian counterpart, i.e., the operator discounts each probability function within the credal set with respect to each of the reliability weight in the set (interval) of such weights. The credal discounting operator has three properties that makes it unique [13]: (1) it can discount any credal set (i.e., it is not restricted to a particular type of credal set), (2) a credal set can be discounted with respect to a set of reliability weights, i.e., one can express reliability imprecisely, and (3) a discounted credal set can be reversed to its original form if the set of reliability weights, used for the discounting, is known.

We also performed two experiments where we evaluate the Bayesian and credal combination operators. In both experiments the sources report credal sets with an implicit uniform second-order distribution as a representation of not favoring any probability function within the sets. For the Bayesian combination operator, we have utilized the expected value of the operand credal sets, i.e., centroids, for obtaining the joint evidence. We evaluated the operators by using a simple score function that gives a reward corresponding to the informativeness

of the joint evidence and a loss according to a specified risk.

In the first experiment, we showed that in scenarios where there exists no risk component, i.e., no cost of reporting an erroneous decision set, it is clearly beneficial to utilize the Bayesian combination operator instead of the credal correspondence. This is true even if one would maintain imprecision by using the credal combination operator and lastly utilize the centroid for decision making. However, this difference in performance was not as clear as in the previous case. Nevertheless, the latter results show that nothing is gained by maintaining imprecision and then using the centroid for constructing the decision set.

By using the result from the experiment, we concluded that if a large risk component is present, the credal method is preferred due to a lower number of erroneous decision sets. However, we introduced a simple cautious Bayesian method, using a single parameter that partitions the probability simplex into regions corresponding to different decision sets, and we showed that such a method can outperform the credal correspondence. One potential problem with the cautious Bayesian method is that one needs to choose an appropriate parameter value. However, we showed that there exist values for which the method outperforms the credal method for a set of risk components.

In essence our results tells us that if there is no risk component in the scenario of interest, then one should use the Bayesian combination operator, even if the sources choose to report imprecision by credal sets. Furthermore, if a risk component does exist in the scenario, then one should use the cautious Bayesian method that we introduced. Hence for both cases it is sufficient to use a single probability function and the Bayesian combination operator for representing respectively combining evidences. From the perspective of computational complexity this is indeed positive results, considering that the number of extreme points of the joint credal evidence in the worst case can grow exponentially with the number of combinations.

The question then is if there exist cases where one might want to maintain imprecision by using the credal combination operator? One possible such scenario could be when there is a human decision maker involved in the scenario, in particular when there exists a risk component. In such cases the decision maker might want to use the credal combination operator in order to maintain imprecision for the purpose of keeping track of worst-case scenarios with respect to the risk.

ACKNOWLEDGMENTS

Computations have been performed with *R* [22]. This work was supported by the Information Fusion Research Program (University of Skövde, Sweden) in partnership with the Swedish Knowledge Foundation under grant 2003/0104 (URL: http://www.infofusion.se).

APPENDIX

THEOREM 1

$$\Phi(\dots\Phi(p_1(X),p_2(X))\dots,p_n(X)) = \frac{p_1(X)\dots p_n(X)}{\sum_{x\in\Omega_X} p_1(x)\dots p_n(x)}. \tag{47}$$

PROOF The proof is by induction. Let us introduce the following shorthand notation

$$p_{1:n}(X) \triangleq \Phi(\Phi(\dots\Phi(p_1(X),p_2(X))\dots,p_{n-1}(X)),p_n(X)). \tag{48}$$

The base case

$$p_{1:2}(X) = \frac{p_1(X)p_2(X)}{\sum_{x\in\Omega_X} p_1(X)p_2(X)} \tag{49}$$

holds by (4). Let the induction hypothesis be

$$p_{1:n-1}(X) = \frac{p_1(X)\dots p_{n-1}(X)}{\sum_{x\in\Omega_X} p_1(x)\dots p_{n-1}(x)}. \tag{50}$$

We need to show that such assumption implies

$$p_{1:n}(X) = \frac{p_1(X)\dots p_n(X)}{\sum_{x\in\Omega_X} p_1(x)\dots p_n(x)}. \tag{51}$$

We have that

$$p_{1:n}(X) = \frac{p_{1:n-1}(X)p_n(X)}{\sum_{x\in\Omega_X} p_{1:n-1}(x)p_n(x)}. \tag{52}$$

By using the induction hypothesis in (50), we get

$$\begin{aligned} p_{1:n}(X) &= \frac{\frac{p_1(X)\dots p_{n-1}(X)}{\sum_{x\in\Omega_X} p_1(x)\dots p_{n-1}(x)}p_n(X)}{\sum_{x\in\Omega_X}\frac{p_1(x)\dots p_{n-1}(x)}{\sum_{x\in\Omega_X} p_1(x)\dots p_{n-1}(x)}p_n(x)} \\ &= \frac{p_1(X)\dots p_n(X)}{\sum_{x\in\Omega_X} p_1(x)\dots p_n(x)}. \end{aligned} \tag{53}$$

By (49)–(53), the proof is complete.

THEOREM 2

$$\Phi_{\mathcal{C}}(\mathcal{P}_1(X),\mathcal{P}_2(X)) = \Phi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}_1(X)),\mathcal{E}(\mathcal{P}_2(X))). \tag{54}$$

PROOF The proof is inspired by Noack, et al. [21, Theorem 2]. First note that

$$\Phi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}_1(X)),\mathcal{E}(\mathcal{P}_2(X))) \subseteq \Phi_{\mathcal{C}}(\mathcal{P}_1(X),\mathcal{P}_2(X)) \tag{55}$$

is trivial. Assume that

$$\Phi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}_1(X)),\mathcal{E}(\mathcal{P}_2(X))) \subset \Phi_{\mathcal{C}}(\mathcal{P}_1(X),\mathcal{P}_2(X)). \tag{56}$$

Then there must exists at least one

$$u(X) \in \Phi_{\mathcal{C}}(\mathcal{P}_1(X),\mathcal{P}_2(X)) \tag{57}$$

such that

$$u(X) \notin \Phi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}_1(X)),\mathcal{E}(\mathcal{P}_2(X))) \tag{58}$$

where $u(X)$ has the following form

$$u(X) = \frac{p_1(X)p_2(X)}{\sum_{x\in\Omega_X} p_1(x)p_2(x)} \tag{59}$$

and where $p_1(X) \in \mathcal{P}_1(X)$ and $p_2(X) \in \mathcal{P}_2(X)$, where at least one of $p_1(X)$ and $p_2(X)$ is not an extreme point. We can express $p_1(X)$ and $p_2(X)$ as

$$p_1(X) = \sum_{i=1}^{m} \lambda_i v_i(X) \tag{60}$$

$$p_2(X) = \sum_{j=1}^{n} \alpha_j w_j(X) \tag{61}$$

where $v_i(X) \in \mathcal{E}(\mathcal{P}_1(X))$, $w_j(X) \in \mathcal{E}(\mathcal{P}_2(X))$, $\lambda_i \geq 0$, $\alpha_j \geq 0$, $1 \leq i \leq m$, $1 \leq j \leq n$, $\sum_{i=1}^{m} \lambda_i = \sum_{j=1}^{n} \alpha_i = 1$, and where there exists at least one $\lambda_i \in (0,1)$ or $\alpha_j \in (0,1)$. By using (60) and (61) in (59), we obtain

$$u(X) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} \lambda_i\alpha_j v_i(X)w_j(X)}{\sum_{x\in\Omega_X}\left(\sum_{i=1}^{m}\sum_{j=1}^{n} \lambda_i\alpha_j v_i(x)w_j(x)\right)}. \tag{62}$$

Let us introduce the following notation

$$\gamma_{i,j} \triangleq \frac{\lambda_i\alpha_j \sum_{x\in\Omega_X} v_i(x)w_j(x)}{\sum_{x\in\Omega_X}\left(\sum_{i=1}^{m}\sum_{j=1}^{n} \lambda_i\alpha_j v_i(x)w_j(x)\right)}. \tag{63}$$

We can now rephrase $u(X)$ as

$$u(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} \gamma_{i,j} \frac{v_i(X)w_j(X)}{\sum_{x\in\Omega_X} v_i(x)w_j(x)}. \tag{64}$$

Since

$$\frac{v_i(X)w_j(X)}{\sum_{x\in\Omega_X} v_i(x)w_j(x)} \in \Phi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}_1(X)), \mathcal{E}(\mathcal{P}_2(X))) \tag{65}$$

and $\gamma_{i,j} \geq 0$, $\sum_{i=1}^{m}\sum_{j=1}^{n} \gamma_{i,j} = 1$, we get (cf (8)):

$$u(X) \in \Phi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}(X)), \mathcal{E}(\mathcal{P}(X))) \tag{66}$$

which is a contradiction.

THEOREM 3

$$\Psi_{\mathcal{C}}(\mathcal{P}(X), \mathcal{W}) = \Psi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}(X)), \mathcal{E}(\mathcal{W})). \tag{67}$$

PROOF First note that

$$\Psi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}(X)), \mathcal{E}(\mathcal{W})) \subseteq \Psi_{\mathcal{C}}(\mathcal{P}(X), \mathcal{W}) \tag{68}$$

is trivial. Assume that

$$\Psi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}(X)), \mathcal{E}(\mathcal{W})) \subset \Psi_{\mathcal{C}}(\mathcal{P}(X), \mathcal{W}). \tag{69}$$

Then there must exists at least one

$$u(X) \in \mathcal{E}(\Psi_{\mathcal{C}}(\mathcal{P}(X), \mathcal{W})) \tag{70}$$

such that

$$u(X) \notin \Psi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}(X)), \mathcal{E}(\mathcal{W})) \tag{71}$$

where $u(X)$ has the following form

$$u(X) = wp(X) + (1-w)p_u(X), \tag{72}$$

where $w \in \mathcal{W}$, and $p(X) \in \mathcal{P}(X)$, and where at least one of w and $p(X)$ is not an extreme point. There are three cases:

Case 1—$p(X) \in \mathcal{E}(\mathcal{P}(X))$, $w \notin \mathcal{E}(\mathcal{W})$: We know that $w = \lambda w_1 + (1-\lambda)w_2$ where $w_1 \neq w_2$, $w_1, w_2 \in \mathcal{E}(\mathcal{W})$, $\lambda \in (0,1)$. We get

$$\begin{aligned}
u(X) &= wp(X) + (1-w)p_u(X) \\
&= p_u(X) + (\lambda w_1 + (1-\lambda)w_2)(p(X) - p_u(X)) \\
&= p_u(X) + \lambda w_1(p(X) - p_u(X)) \\
&\quad + (1-\lambda)w_2(p(X) - p_u(X)) \\
&\quad + \lambda p_u(X) - \lambda p_u(X) \\
&= \lambda(p_u(X) + w_1(p(X) - p_u(X))) \\
&\quad + (1-\lambda)p_u(X) + (1-\lambda)w_2(p(X) - p_u(X)) \\
&= \lambda(p_u(X) + w_1(p(X) - p_u(X))) \\
&\quad + (1-\lambda)(p_u(X) + w_2(p(X) - p_u(X))) \\
&= \lambda(w_1 p(X) + (1-w_1)p_u(X)) \\
&\quad + (1-\lambda)(w_2 p(X) + (1-w_2)p_u(X)).
\end{aligned} \tag{73}$$

Hence, $u(X) \in \Psi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}(X)), \mathcal{E}(\mathcal{W}))$ (cf (8)), which is a contradiction.

Case 2—$p(X) \notin \mathcal{E}(\mathcal{P}(X))$, $w \in \mathcal{E}(\mathcal{W})$: We know that $p(X) = \sum_{i=1}^{n} \alpha_i p_i(X)$, where $p_i(X) \in \mathcal{E}(\mathcal{P}(X))$, $\alpha_i \geq 0$, $\sum_{i=1}^{n} \alpha_i = 1$ where there exists at least one $\alpha_i \in (0,1)$. We get

$$\begin{aligned}
u(X) &= w\left(\sum_{i=1}^{n} \alpha_i p_i(X)\right) + (1-w)p_u(X) \\
&\quad + \left(\sum_{i=1}^{n} \alpha_i(1-w)p_u(X)\right) \\
&\quad - \left(\sum_{i=1}^{n} \alpha_i(1-w)p_u(X)\right) \\
&= \left(\sum_{i=1}^{n} \alpha_i(wp_i(X) + (1-w)p_u(X))\right) \\
&\quad + (1-w)p_u(X) \\
&\quad - \left(\sum_{i=1}^{n} \alpha_i(1-w)p_u(X)\right) \\
&= \sum_{i=1}^{n} \alpha_i(wp_i(X) + (1-w)p_u(X)).
\end{aligned} \tag{74}$$

Hence, $u(X) \in \Psi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}(X)), \mathcal{E}(\mathcal{W}))$ (cf (8)), which is a contradiction.

Case 3—$p(X) \notin \mathcal{E}(\mathcal{P}(X))$, $w \notin \mathcal{E}(\mathcal{W})$: Similar to Case 1 and 2, we have that

$$\begin{aligned}
w &= \lambda w_1 + (1-\lambda)w_2 \\
p(X) &= \sum_{i=1}^{n} \alpha_i p_i(X).
\end{aligned} \tag{75}$$

We get

$$u(X) = (\lambda w_1 + (1-\lambda)w_2)\left(\sum_{i=1}^{n}\alpha_i p_i(X)\right) + (1-(\lambda w_1 + (1-\lambda)w_2))p_u(X). \quad (76)$$

From Case 1 we know that the above equation is equivalent to

$$u(X) = \lambda\left(w_1\left(\sum_{i=1}^{n}\alpha_i p_i(X)\right) + (1-w_1)p_u(X)\right) + (1-\lambda)\left(w_2\left(\sum_{i=1}^{n}\alpha_i p_i(X)\right) + (1-w_2)p_u(X)\right). \quad (77)$$

From Case 2 we know that the above equation is equivalent to

$$u(X) = \lambda\left(\sum_{i=1}^{n}\alpha_i(w_1 p_i(X) + (1-w_1)p_u(X))\right) + (1-\lambda)\left(\sum_{i=1}^{n}\alpha_i(w_2 p_i(X) + (1-w_2)p_u(X))\right). \quad (78)$$

Hence, $u(X) \in \Psi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}(X)), \mathcal{E}(\mathcal{W}))$ (cf (8)), which is a contradiction. Since all possible cases lead to contradictions we must conclude that

$$\Psi_{\mathcal{C}}(\mathcal{P}(X), \mathcal{W}) = \Psi_{\mathcal{C}}(\mathcal{E}(\mathcal{P}(X)), \mathcal{E}(\mathcal{W})). \quad (79)$$

REFERENCES

[1] N. Andréasson, A. Evgrafov, and M. Patriksson
An Introduction to Continuous Optimization.
Studentlitteratur, Lund, Sweden, 2005.

[2] S. Arnborg
Robust Bayesianism: Imprecise and paradoxical reasoning.
In *Proceedings of the 7th International Conference on Information fusion*, June 2004, 407–414.

[3] S. Arnborg
Robust Bayesianism: relation to evidence theory.
Journal of Advances in Information Fusion, **1**, 1 (Apr. 2006), 75–90.

[4] J. O. Berger
An overview of robust Bayesian analysis.
Test, 3 (1994), 5–124.

[5] J. M. Bernardo and A. F. M. Smith
Bayesian Theory.
John Wiley and Sons, 2000.

[6] G. W. Brier
Verification of forecasts expressed in terms of probability.
Monthly Weather Review, 78 (Feb. 1950), 1–3.

[7] I. Couso, S. Moral, and P. Walley
A survey of concepts of independence for imprecise probabilities.
Risk Decision and Policy, 5 (Sept. 2000), 165–181.

[8] F. Cozman
Decision Making Based on Convex Sets of Probability Distributions: Quasi-Bayesian Networks and Outdoor Visual Position Estimation.
Ph.D. thesis, The Robotics Institute, Carnegie Mellon University, 1997.

[9] F. Cozman
A derivation of quasi-Bayesian theory.
Technical Report CMU-RI-TR-97-37, Robotics Institute, Carnegie Mellon University, 1997.

[10] F. G. Cozman
Credal networks.
Artificial Intelligence, **120**, 2 (July 2000), 199–233.

[11] F. G. Cozman
Graphical models for imprecise probabilities.
International Journal of Approximate Reasoning, **39**, 2–3 (June 2005), 167–184.

[12] S. Das
High-Level Data Fusion.
Artech House, 2008.

[13] S. Destercke
A new contextual discounting rule for lower probabilities.
In *Proceedings of International Conference on Information Processing and Management of Uncertainty in Knowledged-Based Systems*, June–July 2010, 198–207.

[14] D. R. Insua and F. Ruggeri, (Eds.)
Robust Bayesian Analysis.
Springer, 2000.

[15] A. Irpino and V. Tontodonato
Cluster reduced interval data using Hausdorff distance.
Computational Statistics, **21**, 2 (2006), 241–288.

[16] A. Karlsson, R. Johansson, and S. F. Andler
On the behavior of the robust Bayesian combination operator and the significance of discounting.
In *Proceedings of the 6th International Symposium on Imprecise Probability: Theories and Applications (ISIPTA)*, 2009.

[17] A. Karlsson, R. Johansson, and S. F. Andler
An empirical comparison of Bayesian and credal set theory for discrete state estimation.
In *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, June–July 2010, 80–89.

[18] A. Karlsson, R. Johansson, and S. F. Andler
An empirical comparison of Bayesian and credal combination operators.
In *Proceedings of The 13th International Conference on Information Fusion*, July 2010.

[19] I. Levi
The Enterprise of Knowledge.
The MIT Press, 1983.

[20] M. E. Liggins, D. L. Hall, and J. Llinas (Eds.)
Multisensor Data Fusion, Second Edition.
CRC Press, 2009.

[21] B. Noack, V. Klumpp, D. Brunn, and U. D. Hanebeck
Nonlinear Bayesian estimation with convex sets of probability densities.
In *Proceedings of the 11th International Conference on Information Fusion*, June–July, 2008, 1–8.

[22] R Development Core Team
R: A Language and Environment for Statistical Computing.
R Foundation for Statistical Computing, Vienna, Austria, 2009.

[23] G. Shafer
A Mathematical Theory of Evidence.
Princeton University Press, 1976.

[24] M. C. Troffaes
Generalizing the conjunction rule for aggregating conflict expert opinions.
International Journal of Intelligent Systems, **21**, 3 (Mar. 2006), 361–380.

[25] P. Walley
Statistical Reasoning with Imprecise Probabilities.
Chapman and Hall, 1991.

[26] P. Walley
Towards a unified theory of imprecise probability.
International Journal of Approximate Reasoning, **24**, 2–3 (May 2000), 125–148.

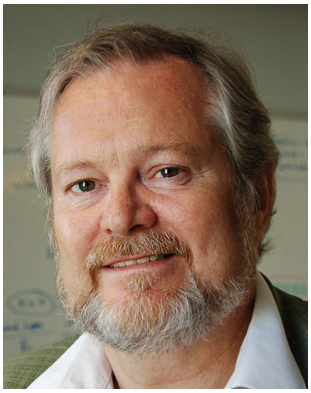
[27] L. A. Zadeh
Review of books: a mathematical theory of evidence.
AI Magazine, **5**, 3 (1984), 81–83.

Alexander Karlsson holds a Ph.D. in computer science at Örebro University, Sweden, 2010, and a M.Sc. in computer science and engineering at Chalmers University of Technology, Sweden, 2004.

Dr. Karlsson is a researcher at the University of Skövde, Sweden. His main research interest is theory for reasoning in the presence of uncertainty.

Ronnie Johansson received his Ph.D. in computer science from the Royal Institute of Technology (KTH), Sweden, in 2006.

Dr. Johansson is a researcher at the Swedish Defence Research Agency (FOI) in Stockholm. He is also a part-time employee at the University of Skövde, Sweden, where he teaches and conducts research. His research has since 2000 focused on autonomous systems and information fusion. He is currently interested in fusion algorithms, management of uncertainty, knowledge representation and information acquisition. He has served on the programme committees of the Information Fusion Conference and the Multisensor Fusion and Integration for Intelligent Systems Conference. In 2000, he spent six months at the RIKEN institute in Saitama, Japan, while working on multi-robot path planning.



Sten F. Andler received his Ph.D. in computer science in 1979 from Carnegie Mellon University, Pittsburgh, PA, and a Ph.D. in computer science from Chalmers University of Technology, Göteborg, Sweden, also in 1979.

He is a Professor of Computer Science at University of Skövde, Sweden, and Program Director of Infofusion, the Research Program in Information Fusion at Skövde. Infofusion is funded for 6+2 years by grants from the Swedish Knowledge Foundation and matching grants from industry and the university, totaling SEK 120+15 million (approx USD $21 million). He has served three years as Dean of Research at the University of Skövde and six years on the Faculty Board at Chalmers University of Technology, Göteborg, Sweden, and is currently serving a second three-year term on the International Society of Information Fusion (ISIF) Board of Directors. He was previously affiliated with the IBM Almaden Research Center and IBM Software Solutions, San Jose, CA, for fourteen years, with brief periods as visiting professor at University of California at Berkeley and Research Intern at Xerox Palo Alto Research Center (PARC). In addition to the Information Fusion Program, now in its seventh year, he heads a research group in Distributed Real-Time Systems (DRTS), with activities in information fusion infrastructures, distributed real-time databases, and model-based software testing.

Dr. Andler's interests are in the areas of information fusion, distributed systems, real-time systems, and databases. He is a Member of the ACM and the IEEE Computer Society. He is a Member of the Editorial Board of Innovations in Systems and Software Engineering.

Detecting a Small Boat using Histogram PMHT

SAMUEL J. DAVEY

The Histogram Probabilistic Multi-Hypothesis Tracker (H-PMHT) is a parametric track before detect algorithm. This article discusses practical issues faced in producing an operationally relevant implementation of the algorithm. The performance of H-PMHT is demonstrated on a high resolution radar example containing a small boat in challenging clutter and compared with a conventional solution based on image segmentation and the Probabilistic Data Association Filter.

Manuscript received November 21, 2010; revised June 19, 2011 and October 19, 2011; released for publication October 26, 2011.

Refereeing of this contribution was handled by Peter Willett

Author's addresses: Defence Science and Technology Organisation, ISRD 200, PO Box 1500, Edinburgh, South Australia 5111, Australia, E-mail: (Samuel.Davey@DSTO.Defence.gov.au); School of Electrical and Electronic Engineering, The University of Adeliade, Australia.

1557-6418/11/$17.00

1. INTRODUCTION

The conventional tracking paradigm is to sequentially apply a single frame detector to each sensor frame and then employ a tracking algorithm to determine which detector outputs originate from targets and which are false alarms. The tracker associates detections from a particular target and estimates parameters of interest for the target [3], [1]. The obvious shortcoming of this approach is that it is impossible for the tracker to recover a target if there are no detector outputs. For low signal strength targets, this implies that the detector threshold must be low and the tracker must attempt to suppress a large number of false alarms. In practice, the tracker can only do so much and for very low signal strength targets, conventional tracking fails [8].

In contrast, track-before-detect (TkBD) algorithms supply the whole predetection sensor frame to the tracker. In essence the tracking problem remains the same, but the measurement function is different. What makes TkBD challenging is that the relationship between the sensor frame and the target state is non-linear and often non-Gaussian, whereas point measurements may often be treated as linear and Gaussian. Apart from a small number of special cases, non-linear non-Gaussian estimation problems cannot be solved with a closed form algorithm. Instead, TkBD algorithms generally use a numerical approximation to make the problem tractable. The numerical approximation may take the form of a fixed discrete grid in state space [17], [2], [21], or a stochastic sampling method may be used (i.e., a particle filter) [19], [4], [18]. Alternatively, the data likelihood ratio may be viewed as an increasing function and the likelihood given a particular state sequence approximated by incoherent or binary integration [14], [20].

The Histogram Probabilistic Multi-Hypothesis Tracker (H-PMHT) is an exception: it is a TkBD algorithm that does not use numerical approximation [22], [24]. H-PMHT uses a unique histogram interpretation of the sensor frame and expectation maximisation (EM) to treat the TkBD problem as one of mixture fitting. The algorithm has the advantage of being capable of handling multiple targets whereas the numerical methods usually assume a single target. In addition, it does not require an assumed statistical distribution for the amplitude of the scene background, which may be difficult to adequately approximate in a realistic application.

Despite its advantages, little work has been published on H-PMHT besides the original algorithm development [22], [24] and its extension to spectral data [23]. Pakfiliz and Efe presented marginalised expressions for a two-dimensional filtering problem and compared H-PMHT with a conventional tracking approach (Interacting Multiple Model Probabilistic Data Association with Amplitude Information) [16]. They showed that H-PMHT provided lower estimation errors. However, the comparison used two independent simulated

data sets rather than producing the point-measurements for the conventional tracker from the sensor frame provided to H-PMHT. Also, the issue of track initiation, which is essentially the whole purpose of TkBD, was not considered. Davey, et al. [8] compared the initiation performance of H-PMHT with other TkBD methods and with sequential detection and tracking using simple straight line scenarios. The comparison showed that H-PMHT provided detection and estimation performance similar to the other TkBD approaches considered but at a fraction of the computation cost. While these results provide strong motivation for using H-PMHT, the scenarios were simplistic and not operationally relevant. Also, only limited implementation information was provided. Recently, Wieneke used a Wishart prior to enable the H-PMHT to estimate the covariance matrix of the target signature [26] and Davey demonstrated how a particle filter could be used for target state estimates when the target signature is highly non-Gaussian [6].

This article builds on the comparisons already available and addresses whether the algorithm still performs well with real sensor data, which inevitably does not match the modeling assumptions. It shows that the advantages illustrated on thumbnail images in [8] do extrapolate to larger, more relevant image sizes. The performance of the algorithm is investigated as a function of the image size, the number of targets and of the dynamic range in target echo strength. A matrix-vector form of the algorithm is derived for the case of a two dimensional sensor image frame and this is shown to significantly improve implementation efficiency. An efficient method for incorporating non-Gaussian targets is developed and compared with the computationally demanding particle filter approach introduced in [6]. Finally the H-PMHT is then demonstrated on an experimental data set containing a small boat collected by Defence Research and Development Canada (DRDC) [13]. The performance of H-PMHT on the experimental data is compared with a conventional tracking approach consisting of an image segmentation detector and two alternate trackers: a Probabilistic Data Association Filter (PDAF) and a point-measurement PMHT. The main purpose of this example is not a quantitative comparison, but rather to qualitatively investigate the scaling performance of the algorithm to high resolution radar imagery and the sensitivity of the algorithm to mismatch in the target and clutter models.

The article is arranged as follows: a review of the H-PMHT is presented in Section 2; Section 3 illustrates the performance scaling of the algorithm as a function of the measurement image volume, number of targets and diversity of targets, and derives an efficient implementation representation for two dimensional sensor images; Section 4 presents a track management method for H-PMHT; Section 5 introduces a new method for non-Gaussian targets; and Section 6 compares the performance of H-PMHT and conventional tracking on a challenging maritime surveillance experiment.

2. REVIEW OF H-PMHT

The H-PMHT algorithm, as introduced in [22], is now reviewed. H-PMHT is derived by interpreting the sensor image as a histogram of observations of an underlying random process. The received energy in each cell is quantised, and the resulting integer is treated as a count of the number of measurements that fell within that cell. The sum over all of the cells is the total number of measurements taken. The probability mass function for these discrete measurements is modelled as a multinomial distribution where the probability mass for each cell is the superposition of target and noise contributions.

The probability that an individual histogram shot falls in cell l is

$$\frac{P_l(\mathbf{X}_t)}{P(\mathbf{X}_t)} \tag{1}$$

where

$$P(\mathbf{X}_t) \equiv \sum_l P_l(\mathbf{X}_t) \tag{2}$$

and

$$P_l(\mathbf{X}_t) = \int_{B_l} f(\tau \mid \mathbf{X}_t) d\tau. \tag{3}$$

The spatial extent of cell l (which is of arbitrary dimensionality) is B_l and $\mathbf{X}_t$ is the system state vector at time t, i.e., it summarises all of the targets.

The underlying continuous spatial density, $f(\tau \mid \mathbf{X}_t)$ is the superposition of a background clutter model and M target components

$$f(\tau \mid \mathbf{X}_t) \equiv f(\tau \mid \mathbf{X}_t; \Pi_t) = \pi_t^0 G_0(\tau) + \sum_{m=1}^{M} \pi_t^m G^m(\tau \mid \mathbf{x}_t^m) \tag{4}$$

where $\mathbf{x}_t^m$ is the state of the mth target and the mixing proportions form a probability vector, i.e., $\pi_t^m \geq 0$ and

$$\sum_{m=0}^{M} \pi_t^m = 1. \tag{5}$$

The mixing proportions can be interpreted as the relative power of each target. In the simplest case, the background clutter model is uniform and $G_0(\tau)$ is a constant. In spatially non-uniform clutter, mapping approaches such as [5], [10] may be used. The target component $G^m(\tau \mid \mathbf{x}_t^m)$ may be common across targets, e.g. it could represent the point spread function (psf) for a sensor observing point scatterers, or each target may have a unique signature, such as with high resolution optical sensors. Either way, it is assumed to have a known functional form with potentially unknown parameters.

H-PMHT is an EM algorithm that treats the assignment of histogram shots to model components and the precise location of shots as missing data. In addition, it allows for unobserved cells. These are notionally sensor pixels for which no data was collected. One use for this concept is in tracking targets near the edge of the sensor frame [24]. The data from these unobserved

cells is also treated as missing data. Assuming an existing estimate of the system state, $\mathbf{X}_t^{(i)}$, and the mixing proportions, $\Pi_t^{(i)}$, H-PMHT determines the probability of the missing data (E-step) and then refines the state estimate (M-step).

Denote the per-cell proportion of the contribution of target m at time t as

$$P_l^m(\mathbf{x}_t^{m(i)}) = \int_{B_l} G^m(\tau \mid x_t^{m(i)}) d\tau \tag{6}$$

then

$$P_l(\mathbf{X}_t^{(i)}) = \sum_{m=0}^{M} \pi_t^{m(i)} P_l^m(\mathbf{x}_t^{m(i)}). \tag{7}$$

The observed power in cell l at time t is denoted z_{tl}. Define

$$\|Z_t\| = \sum_l z_{tl} \tag{8}$$

namely the L_1 norm, and let

$$\bar{z}_{tl} = \begin{cases} \dfrac{z_{tl}}{P_l(\mathbf{X}_t^{(i)})} & l \in \mathcal{L} \\ \dfrac{\|Z_t\|}{P(\mathbf{X}_t^{(i)})} & l \in \bar{\mathcal{L}} \end{cases} \tag{9}$$

where $\mathcal{L}$ is the set of all observed cells and $\bar{\mathcal{L}}$ is the set of all unobserved cells, which may be empty. $\mathcal{S}$ is the union of sets $\mathcal{L}$ and $\bar{\mathcal{L}}$.

The parameters to be estimated are the mixing proportions, π_t^m and the target states, $\mathbf{x}_t^m$. The updated mixing proportion estimate is given by

$$\pi_t^{m(i+1)} = \frac{p_t^{m(i)}}{\sum_{m=0}^{M} p_t^{m(i)}} \tag{10}$$

where

$$p_t^{m(i)} = \pi_t^{m(i)} \sum_{l \in \mathcal{S}} \bar{z}_{tl} P_l^m(x_t^{m(i)}). \tag{11}$$

The state estimate requires the maximisation of the function

$$Q_{mX} = \sum_{t=1}^{T} \frac{\|Z_t\|}{P(\mathbf{X}_t^{(i)})} \log\{p(\mathbf{x}_t^m \mid \mathbf{x}_{t-1}^m)\} + \sum_{t=1}^{T} \sum_{l \in \mathcal{S}} \pi_t^{m(i)} \bar{z}_{tl} \int_{B_l} G^m(\tau \mid x_t^{m(i)}) \log\{G^m(\tau \mid x_t^m)\} d\tau. \tag{12}$$

For the case of linear Gaussian statistics, Streit [22] demonstrated that this maximisation problem is equivalent to a point measurement filtering problem with synthetic measurements

$$\tilde{z}_t^{m(i)} = \frac{\pi_t^{m(i)}}{p_t^{m(i)}} \sum_{l \in \mathcal{S}} \bar{z}_{tl} P_l^m(x_t^{m(i)}) \tilde{z}_{tl}^{m(i)} \tag{13}$$

where the *cell-level centroid*, $\tilde{z}_t^{m(i)}$, is given by

$$\tilde{z}_{tl}^{m(i)} = \frac{1}{P_l^m(x_t^{m(i)})} \int_{B_l} \tau G^m(\tau \mid x_t^{m(i)}) d\tau. \tag{14}$$

The associated synthetic measurement covariance is

$$\tilde{R}_t^m = \frac{1}{p_t^{m(i)}} R_t^m \tag{15}$$

and the synthetic process covariance is

$$\tilde{Q}_t^m = \frac{P(X_t^i)}{\|Z_t\|} Q_t^m. \tag{16}$$

A Kalman Filter can be used to solve this point measurement filtering problem.

The H-PMHT algorithm then consists of iteratively equating:

- cell probabilities, (6) and (7)
- expected measurements, (9)
- cell-level centroids, (14)
- synthetic measurements and covariance matrices, (13), (15) and (16)
- mixing proportion estimates, (10)
- target state estimates, by filtering the synthetic measurements and covariances

3. EFFICIENT IMPLEMENTATION

The majority of the H-PMHT literature does not discuss the computation resource required for the algorithm. The exception is the comparison work in [8], which shows that the algorithm is inexpensive compared with other TkBD methods on thumbnail images. However, it is intuitive that a direct implementation of the equations reviewed in the previous section will have a computation cost that scales linearly with the number of pixels in the data image. This is consistent with results reported by Vo, et al. [25] where a non-linear growth was found as a function of resolution (the authors of [25] suggest that this growth is exponential, although one should expect it to be quadratic). This growth is clearly undesirable and potentially disastrous for video imagery, which can be expected to contain millions of pixels.

We now demonstrate that with a little finesse, this quadratic complexity can be tamed, and introduce other measures aimed at improved efficiency.

3.1. Pixel Gating

The standard H-PMHT equations as presented in Section 2 show a linear complexity in the number of targets and the number of pixels in the sensor image. Put simply, there are a number of operations which (at first blush) should be performed for each target over every pixel. However, intuition suggests that pixels very distant from the target state estimate will have negligible influence.

This is nothing more than an expression of the ubiquitous method of gating for point-measurement estimators. For example, under Probabilistic Data Association or point-measurement PMHT, one could calculate an association probability for every measurement in the

frame, but many of these measurements are so far away from the target that we know their association probability will be zero: there is no need to waste the computer's time in calculating these numbers.

To make this discussion more crisp, let $\mathcal{G}^m$ be the set of cells with non-negligible probability for target m, namely $\mathcal{G}^m \equiv \{l : P_l^m(x_t^m) > \epsilon\}$ for some arbitrary small ϵ. Equation (11) can then be replaced with

$$p_t^{m(i)} = \pi_t^{m(i)} \sum_{l \in \mathcal{G}^m} \bar{z}_{tl} P_l^m(x_t^{m(i)}) \tag{17}$$

and similarly for (13).

It is clear that for a fixed target, the cardinality of $\mathcal{G}^m$ is independent of the cardinality of $\mathcal{S}$, ignoring edge effects. That is, the complexity is independent of the image size for a fixed target size.

3.2. Clustering

Another common method for reducing complexity in multi-target trackers is clustering [3]. For algorithms whose computation complexity is more than linear in the number of targets, it makes sense to divide the total set of targets into subsets of potentially interacting targets such that each subset is independent of the others. These subsets are commonly called clusters and the process of forming them clustering. For H-PMHT the complexity is linear in the number of targets, so there would seem to be no real advantage in performing clustering.

While clustering will not improve the computation cost for H-PMHT, it will make a difference to the memory requirements. Assuming the gating approach described above, there will be a collection of gates, $\mathcal{G}^m$. Each track only needs to operate over the pixels in its own gate. However, there are some actions that operate over the whole image. Specifically, the overall cell probability $P_l(\mathbf{X}_t)$ should be determined for the region $\bigcup \mathcal{G}^m$, and some of the track management functions to follow operate over the whole image. If clustering is used, then the amount of memory required for large arrays would be reduced. The performance improvement gained can be significant for large images where memory caching plays an important role in overall speed.

Clustering methods are well known in the context of point-measurement multi-target tracking and are not pursued in this paper.

3.3. Partial EM Update

H-PMHT is an EM algorithm that iterates data association and state estimation stages until convergence. The proper way to measure this convergence is through the EM auxiliary function, which can be expressed as

$$Q(\mathbf{X}, \Pi \mid \mathbf{X}^{(i)}, \Pi^{(i)}) = \sum_{t=1}^{T} Q_{t\Pi} + \sum_{m=0}^{M} Q_X^m \tag{18}$$

where the term $Q_{t\Pi}$ depends only on the mixing proportions, and each term Q_X^m depend on the state sequence of a single target [22]. Importantly, the target state sequences are not coupled, so each Q_X^m can be independently optimised and the estimation stage is a bank of parallel single target estimators.

EM iterations should be repeated until (18) converges, i.e., all target states and the mixing proportions are estimated repeatedly until convergence. The $Q_{t\Pi}$ term couples all the targets together through normalisation, so this term will not converge until all of the target states converge. However, it is likely that each target will converge at a different rate. In particular, when a track has been spuriously initiated without the support of a real target, the state estimates will usually converge very quickly, since the data association stage finds no support for the track and it essentially dead-reckons. In contrast, tracks that are initiated on a real target response may take many iterations to recover from initialisation error, particularly if the target velocity is high. So it is likely that much effort would be spent repeating the estimation of target states for some tracks that have already converged while the algorithm waits for other tracks that have not converged.

An intuitive way to reduce the redundant reestimation of converged states is to use partial EM steps. Under a partial EM step, only a subset of the states are iterated. Because of the factorised form of the auxiliary function (18), the only way that a change in the state of target m will effect the target auxiliary component for a different target p, Q_X^p, is if the change in the state of target m affects the data association stage for target p. Under the assumption that this will not happen if target p has already converged, then one may employ a partial EM iteration that keeps the state of p fixed and refines the state of m. This not only saves on the computation required for the estimator of p, but also the computation required for the cell probabilities, expected measurements, centroids and synthetic measurements. Adopting partial EM iterations in this manner can reduce the overall algorithm resource requirement by as much as an order of magnitude because the majority of the tracks carried by the algorithm are spurious and converge very quickly.

In practice the convergence test for the algorithm will limit the number of iterations based on the maximum tolerable processing delay per frame. By using partial updates, the time taken for each iteration is reduced as the iteration count increases, since fewer tracks still require refinement. The result is that a much greater maximum number of iterations may be used without significant impact on the overall algorithm speed.

The partial-EM H-PMHT consists of the following steps:

- For converged tracks, fix the cell probabilities using the value determined on the final iteration before convergence.

- For non-converged tracks, determine the new cell probabilities for this iteration using (6).
- Calculate the overall cell probabilities (7).
- For non-converged tracks determine:
- expected measurements, (9)
- cell-level centroids, (14)
- synthetic measurements and covariance matrices, (13), (15) and (16)
- target state estimates, by filtering the synthetic measurements and covariances
- For all tracks, determine new mixing proportion estimates, (10)

In principle the convergence test should be based on the cost function element Q_X^m. However, the only reason to determine this quantity is to test for convergence, and this is relatively expensive. Therefore the implementation used in this paper used a convergence test based on the target state: when the difference between the state estimates for consecutive iterations dropped below a threshold, the track was deemed converged. A maximum of ten iterations was also applied.

3.4. Separable Measurement Function

The H-PMHT derivation uses single indexing of the sensor cells. Single indexing does not limit the sensor dimensionality, but it makes it more difficult to decouple the sensor dimensions if they are independent. For the following, assume a two dimensional sensor with an independent grid such that

$$\int_{B_l} G(\tau \mid \mathbf{x}) d\tau \equiv \int_{B_X^i} \int_{B_Y^j} G(u, v \mid \mathbf{x}) dv\, du \tag{19}$$

where the arbitrary index l corresponds to the same cell as the arbitrary double index i, j.

Assume also a separable point spread function:

$$G(u, v \mid \mathbf{x}) \equiv g_X(u \mid \mathbf{x}) g_Y(v \mid \mathbf{x}). \tag{20}$$

The standard H-PMHT equations presented in the previous section can solve this problem, but it's desirable to exploit the structure of the target function to achieve a factorised algorithm. The results derived here are equivalent to the algorithm presented in [16]. Here more detail is shown to justify the end result and compact vector notation is adopted to help make compact intuitive expressions and to point towards efficient implementation.

In the following discussion, the time subscripts and iteration indices are suppressed to simplify notation.

The per-cell target probabilities (6) directly factorise into a product of two integrals:

$$\begin{aligned} P_{ij}^m(\mathbf{x}_t^m) &= \left\{ \int_{B_X^i} g_X(u \mid \mathbf{x}^m) du \right\} \left\{ \int_{B_Y^j} g_Y(v \mid \mathbf{x}^m) dv \right\} \\ &\equiv P_X^m(i) P_Y^m(j). \end{aligned} \tag{21}$$

Define the stacked vectors

$$\mathbf{P}_X^m = [P_X^m(1), P_X^m(2), \ldots P_X^m(N_X)]^{\mathrm{T}} \tag{22}$$

where N_X is the number of cells in the X-dimension. Similarly define $\mathbf{P}_Y^m$.

Let P_X be a matrix of the X per-cell contributions

$$\mathsf{P}_X = [\mathbf{P}_X^0, \mathbf{P}_X^1, \ldots \mathbf{P}_X^M] \tag{23}$$

and similarly define P_Y, then

$$\begin{aligned} \mathsf{P} \equiv \{P_{ij}(\mathbf{X})\} &= \left\{ \sum_{m=0}^{M} \pi^m P_{ij}^m(\mathbf{x}^m) \right\} \\ &= \mathsf{P}_X \Lambda \mathsf{P}_Y^{\mathrm{T}} \end{aligned} \tag{24}$$

where Λ is a diagonal matrix of the π^m. Note that (24) is simply a matrix-vector version of (7). They are equivalent, but (24) represents an explicit factorisation of (7) exploiting the separable point spread function.

The normalised measurements are

$$\bar{\mathsf{Z}} = \mathsf{Z}_t ./ \mathsf{P} \tag{25}$$

where the Matlab-style ./ notation denotes element-wise division.

The unscaled mixing proportion estimate is given by

$$p^m = \pi^m \sum_{i,j} \bar{z}_{ij} P_X^m(i) P_Y^m(j) = \pi^m \{\mathbf{P}_X^m\}^{\mathrm{T}} \bar{\mathsf{Z}} \mathbf{P}_Y^m. \tag{26}$$

Again, (26) is equivalent to (11) but explicitly factorises the calculation of p^m due to the separable point spread function.

The cell-level centroids are given by

$$\begin{aligned} \tilde{\mathbf{z}}_{ij}^m &= \frac{1}{P_X^m(i) P_Y^m(j)} \int_{B_X^i} \int_{B_Y^j} \begin{bmatrix} u \\ v \end{bmatrix} g_X(u \mid \mathbf{x}^m) g_Y(v \mid \mathbf{x}^m) dv\, du, \\ &= \frac{1}{P_X^m(i) P_Y^m(j)} \begin{bmatrix} \int_{B_X^i} \int_{B_Y^j} u g_X(u \mid \mathbf{x}^m) g_Y(v \mid \mathbf{x}_t^m) dv\, du \\ \int_{B_X^i} \int_{B_Y^j} v g_X(u \mid \mathbf{x}^m) g_Y(v \mid \mathbf{x}_t^m) dv\, du \end{bmatrix} \\ &= \frac{1}{P_X^m(i) P_Y^m(j)} \begin{bmatrix} P_Y^m(j) \int_{B_X^i} u g_X(u \mid \mathbf{x}^m) du \\ P_X^m(i) \int_{B_Y^j} v g_Y(v \mid \mathbf{x}^m) dv \end{bmatrix} \\ &= \begin{bmatrix} 1/P_X^m(i) \int_{B_X^i} u g_X(u \mid \mathbf{x}^m) du \\ 1/P_Y^m(j) \int_{B_Y^j} v g_Y(v \mid \mathbf{x}^m) dv \end{bmatrix} \equiv \begin{bmatrix} \tilde{z}_X^m(i)/P_X^m(i) \\ \tilde{z}_Y^m(j)/P_Y^m(j) \end{bmatrix} \end{aligned} \tag{27}$$

i.e., as intuitively expected, the centroids are independent in the sensing dimensions. Note that the term $\tilde{z}$ is different to (14) in that it does not include the normalising term (abusing notation, $\tilde{z}_{14}^m = \tilde{z}_{27}^m / P^m$). This has been done to achieve a more efficient vector formulation in the following step.

Thus the synthetic measurements are

$$\begin{aligned}\tilde{\mathbf{z}}^m &= \frac{\pi^m}{p^m}\sum_i\sum_j \bar{z}_{ij}P_X^m(i)P_Y^m(j)\tilde{\mathbf{z}}_{ij}^m,\\ &= \frac{\pi^m}{p^m}\begin{bmatrix}\sum_i\sum_j \bar{z}_{ij}P_X^m(i)P_Y^m(j)\tilde{z}_X^m(i)/P_X^m(i)\\ \sum_i\sum_j \bar{z}_{ij}P_X^m(i)P_Y^m(j)\tilde{z}_Y^m(j)/P_Y^m(j)\end{bmatrix}\\ &= \frac{\pi^m}{p^m}\begin{bmatrix}\sum_i \tilde{z}_X^m(i)\left\{\sum_j \bar{z}_{ij}P_Y^m(j)\right\}\\ \sum_j \tilde{z}_Y^m(j)\left\{\sum_i \bar{z}_{ij}P_X^m(i)\right\}\end{bmatrix}\\ &= \frac{\pi^m}{p^m}\begin{bmatrix}\{\tilde{\mathbf{z}}_X^m\}^{\mathrm{T}}\bar{\mathbf{Z}}\mathbf{P}_Y^m\\ \{\mathbf{P}_X^m\}^{\mathrm{T}}\bar{\mathbf{Z}}\tilde{\mathbf{z}}_Y^m\end{bmatrix}\end{aligned} \tag{28}$$

where

$$\tilde{\mathbf{z}}_X^m = [\tilde{z}_X^m(1),\ldots\tilde{z}_X^m(N_X)]^{\mathrm{T}} \tag{29}$$

and similarly for $\tilde{\mathbf{z}}_Y^m$.

The synthetic measurement and process covariances remain the same, and the state estimates may now be determined using a Kalman Filter.

The results above can be extended to higher dimensions, for example if the sensor forms range, Doppler, azimuth and elevation bins, then the sensor image would be four dimensional. In such a case, compact expressions such as (28) rely on stacking the dimensions to take advantage of matrix algebra.

The matrix-vector formulation above does not explicitly reduce the number of operations required. However, if the algorithm is implemented in a higher level computer language, it is likely that expressing the mathematics this way makes the algorithm amenable to the language's intrinsic pipelining and parallelisation capabilities. It is the author's experience that within the Matlab environment the speed difference is significant.

3.5. Numerical Illustration

Consider a simplistic example to numerically verify the scaling of the algorithm as a function of image size and the number of targets. At this stage we have not introduced track management, so the algorithm is initialised with the true state of each target and recursively updated using the various efficiency measures described above. Fig. 1 shows an example scenario with 24 targets. All of the targets move with a speed of 1 pixel per frame and are positioned away from the image boundaries and each other. For small images with many targets, there will be inevitable overlap.

Monte Carlo trials of the scenario in Fig. 1 were performed as a function of the number of targets and the image size. The target spacing was proportional to the image size. For each combination of target count and image size, 100 Monte Carlo trials were performed (a total of $9 \times 16 \times 100 = 14400$ trials). The CPU time spent by H-PMHT is shown in Fig. 2 with 1-sigma error-bars. Fig. 2(a) shows the increase in cost as a function of the number of targets with a separate curve

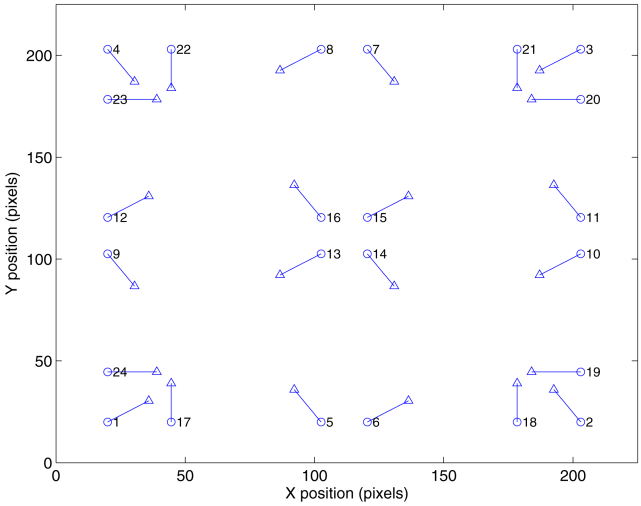


Fig. 1. Scaling scenario.

for each image size. There is no statistical difference between any of these curves. The growth appears to be somewhat less than linear in the number of targets. This is because of the pipelining effect mentioned in the previous section: the code itself has a simple loop over the targets, but the cost of executing this loop is less than linear because of loop overheads and the Matlab interpreter's ability to optimise the loop. The independence of the CPU cost from total image size is emphasised in Fig. 2(b) where the cost for a fixed number of targets is shown as a function of image size. Different curves show an increasing target density, but each curve remains flat.

Next the effect of partial EM iterations was investigated using a fixed number of 4 targets and an image size of 50,000 pixels. The target geometry was the same as in the scaling simulations. However, track management was used to automatically acquire the targets instead of initialising with the truth, as above. The track manager is described in Section 4. The maximum number of EM iterations for a single time update was limited to 100, but this limit was rarely met. The number of iterations for each update of each track was counted for 100 monte carlo trials. Under full EM iterations, all tracks are iterated until all tracks have converged, so the number of iterations was the same for all tracks at a particular scan. Under partial EM iterations, iterations were only performed on tracks yet to converge, so the number of iterations can be different for each track.

The track manager used candidate and established tracks as part of the initiation scheme, as described in Section 4. Separate counts were recorded for candidate and established tracks. The mean number of EM iterations for established tracks was 2.7 under full EM iterations and 2.1 under partial EM iterations. The benefit is minor because the uncertainty in the target state is small and only a couple of EM iterations are required. In effect, the prior is very good because these tracks correspond to real targets that have been under track

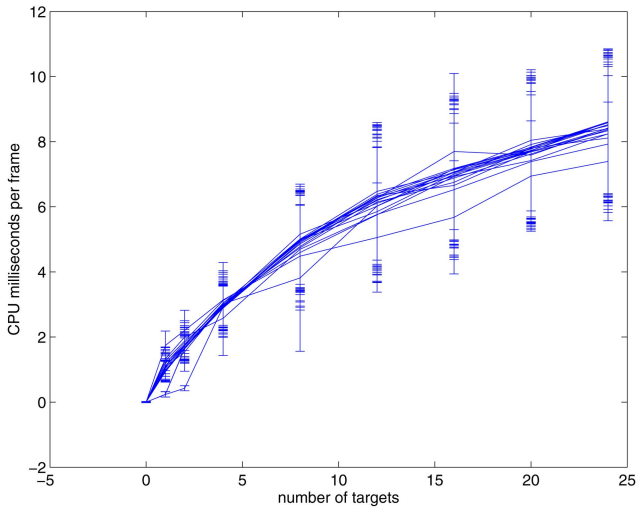


Fig. 2. CPU requirement scaling. (a) Growth with number of targets. (b) Growth with image size.

for some time. In contrast, the mean number of EM iterations for candidate tracks was 55.8 under full EM and 12.0 under partial EM. Histograms of the number of iterations per track under full and partial EM are shown in Fig. 3.

In this example, the tracker only formed established tracks on the 4 targets whereas there were an average of 74.8 candidate tracks. This means that the computational performance is limited by the candidate tracks, and the partial EM iterations have delivered roughly a factor of 5 improvement. The number of candidate tracks depends on the size of the surveillance region and, in practice, the maximum number of iterations would be limited to many fewer than 100. This means that the benefit of partial EM iterations will be application dependent and likely much less than the factor of 5 here. Nevertheless, it offers a substantial improvement.

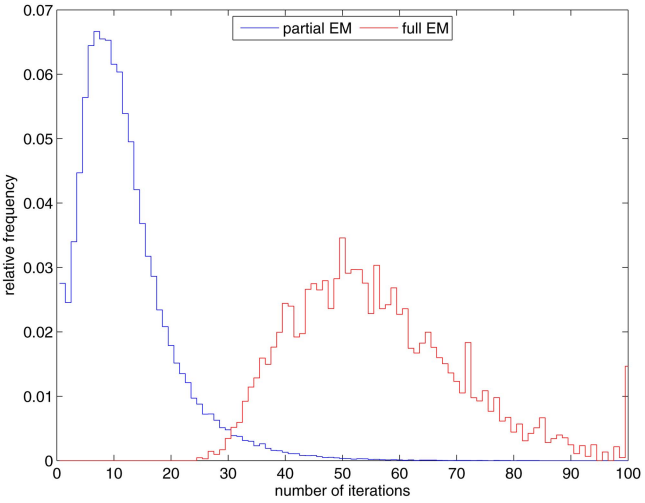


Fig. 3. Benefit of partial EM iterations.

4. TRACK MANAGEMENT

The core H-PMHT algorithm updates existing tracks, but does not provide a means for initiating new tracks or terminating old tracks. These functions are essential for a operationally relevant tracker, especially since the benefit of H-PMHT is espoused as high detection sensitivity. This section describes how track initiation and termination can be incorporated into the algorithm.

To begin with, observe that the H-PMHT as a black-box looks very similar to a Joint Probabilistic Data Association tracker: there is a list of tracks, grouped into clusters, which interact with each other, and the algorithm modifies these tracks based on observed data. Given this structural familiarity, it should come as no surprise that well known track maintenance methods can be directly used with H-PMHT.

The tracker uses two lists of tracks: established tracks, which the algorithm has high confidence in; and candidate tracks (also referred to as tentative tracks), which correspond to potential targets, but do not have a high confidence. Candidate tracks are promoted to established based on their confidence, and track termination is also based on confidence. Established tracks have priority access to the sensor data, followed by candidate tracks, and finally a process which forms new candidate tracks, as shown in Fig. 4. In the figure, the two track-update blocks are separate instances of the H-PMHT algorithm. For the purposes of this paper they are identical, although for some applications it may be appropriate to vary their parameters. The measurements and residual measurements are sensor images. The hierarchial data access prevents the tracker from forming duplicate tracks on targets that already have an existing track in either the established or candidate list.

Recall that these two track lists are a representation of a mixture model. In this light, one can view the established track list as those components of the mixture that are persistent and relatively strong, whereas the

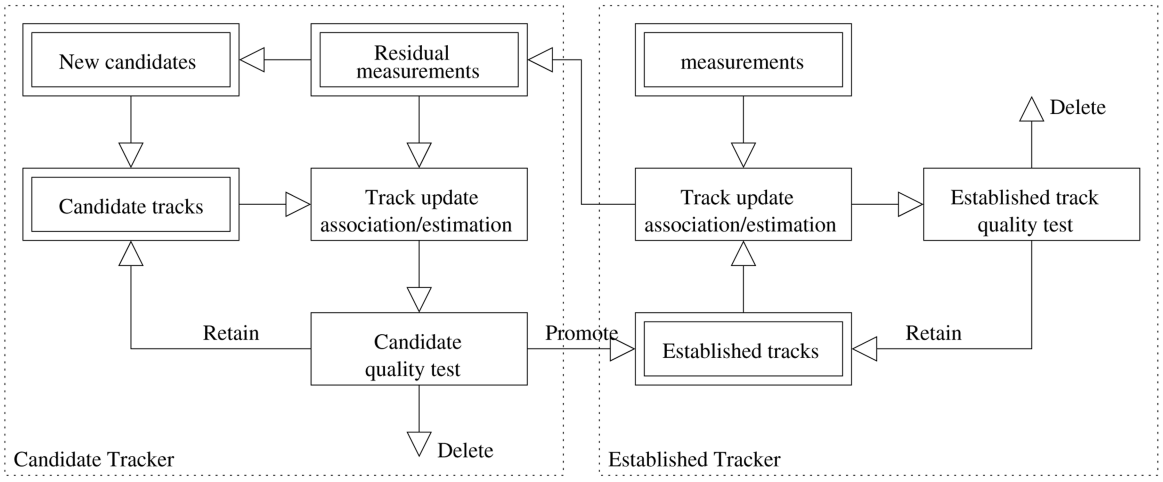


Fig. 4. Track management flow diagram.

candidate track list is a mixture representation of the remaining image structure that is inconsistent with the clutter model.

In order to realise the management structure in Fig. 4 the following elements are required: a measure of track quality or confidence; a set of rules for track decisions; a method for vetting the sensor image to enable hierarchial data access; and a method for forming new candidate tracks based on data. Each of these is now described.

The quality of each track is quantified by estimating the track's SNR. In this context, define the SNR as the ratio of the peak level of the target spread function to the local noise floor. It will be expressed in decibels, namely

$$s_t^m \equiv 10\log_{10}\frac{\pi_t^m \max_l\{P_l^m\}}{\pi_t^0}. \tag{30}$$

For a rectangular image, a Gaussian point spread function and uniform clutter, the SNR is simplified to

$$s_t^m = 10\log_{10}\frac{\pi_t^m N_X N_Y}{\pi_t^0 |2\pi\mathsf{R}|}. \tag{31}$$

The SNR equations above use the true mixing proportions, π_t^m, which must be replaced by their estimates, since they are unknown. This paper is focused on a sequential implementation of H-PMHT, which means that these estimates are based on a single frame of data and are therefore expected to fluctuate. An alternative is to introduce dynamics within the π_t^m priors through the Hysteresis approach [7]. This method has not yet been applied to H-PMHT. However, the algorithm is very similar to the point-measurement PMHT used in [7] and the application of the Hysteresis model would be relatively straightforward.

The next element of the track manager is the set of decision rules. Since the track confidence is driven by SNR estimates based on a single frame of data, the track quality statistics are expected to fluctuate and it is appropriate to use M out of N style decisions, much the same as might be used in a conventional point-measurement tracker. The particular threshold and suitable values for M and N are application dependent. The values used in specific examples in this paper are provided with the examples.

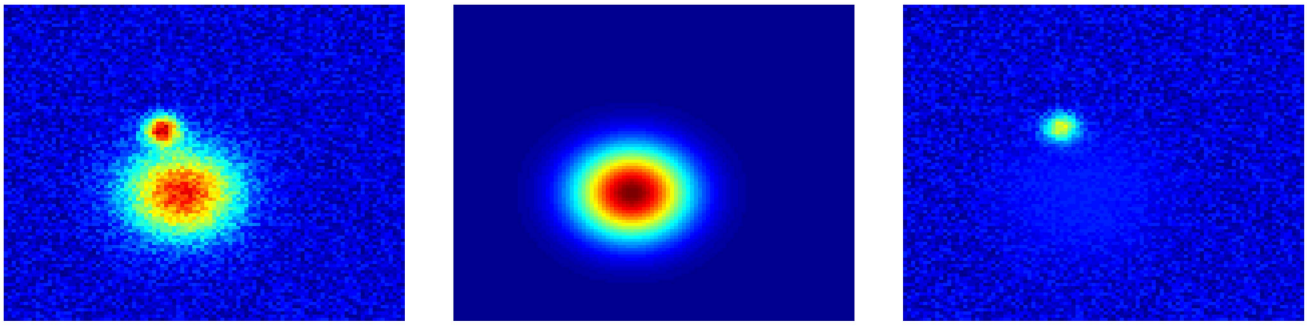
The data vetting function is used to give the different status tracks hierarchial access to the sensor image. In a point-measurement tracker, this is as simple as removing measurements that have been associated with established tracks from the measurement list before data association with the candidate tracks. The H-PMHT requires a vetting method that modifies the sensor image and removes target energy associated with existing tracks. For this, the track manager uses the *whitening* method proposed in [24].

The H-PMHT estimates the parameters of a mixture distribution. On convergence, the total cell probabilities, P, represent the fitted distribution. Near targets, the P_l values will be relatively high and for pixels far from targets, the P_l value will be purely the background noise component, $\pi^0 P_l^0$. The ratio of the total cell probability to the background component, $P_l/(\pi_t^0 P_l^0)$ is a mask which is unity far from targets and has a higher value close to targets. Dividing the sensor image pixel-wise by this mask suppresses the targets

$$z'_{tl} = \frac{\pi_t^0 P_l^0}{P_l} z_{tl}. \tag{32}$$

This whitened image is provided as input to subsequent stages. Namely, the established track mixture whitens the image before candidate tracks are updated and then the candidate track mixture whitens the image before the manager attempts to start new candidate tracks.

Fig. 5 shows a simplified illustration of the process. In the example, there are two targets: one broad target at $(45,60)$ and a more compact one at $(40,40)$. The background noise is uniform and Gaussian. Fig. 5(a) shows the sensor image. Suppose that the tracker has an established track on the broad target. Fig. 5(b) shows the mixture model corresponding to the established track



(a) (b) (c)

Fig. 5. Sensor image vetting. (a) Original sensor image. (b) Mixture pdf of existing tracks. (c) Whitened sensor image.

list. Fig. 5(c) shows the whitened image achieved by pixel-wise dividing the image in (a) by the image in (b). The second target is preserved but the first target, for which a track already exists, is suppressed.

The final element of the track manager is the process to create new candidate tracks. The author has tested two alternatives for this process. The first alternative is to use incoherent integration along with the process model to accumulate data over multiple frames. When a pixel exceeds a prescribed threshold, a new candidate track is created in that location with zero velocity. This is similar to one-point initialisation [3] but the detector input is an accumulated image instead of a single frame. The accumulated image is recursively defined as

$$\zeta_t = (1-\alpha)\mathsf{Z}_t + \alpha\{\zeta_{t-1} \star \mathsf{T}\} \tag{33}$$

where $0 < \alpha < 1$ is a forgetting factor, $\star$ is a two dimensional convolution, and T is a transition kernal which gives the prior probability that a target in pixel (i, j) at time $t-1$ will be in pixel (i', j') at time t: it is specified by the prior target velocity distribution. The author found this method to work well for slow moving targets.

The application in this paper uses the second method for forming candidates: two point differencing [3]. Here a single frame detector is used to locate potential targets and then a new candidate track is formed on pairs of detections from consecutive frames when they are sufficiently close. This was found to work better in the application because the target moves through tens of pixels between each frame and so the transition kernal T above was too broad: the accumulated image was dominated by the current measurement Z_t and too many candidate tracks were initiated on clutter.

5. STRUCTURED TARGETS

The original presentation of the H-PMHT algorithm [22] showed that for linear Gaussian mixture components, i.e., $G(\tau \mid \mathbf{x}) = \mathcal{N}(\tau; \mathsf{H}\mathbf{x}, \mathsf{R})$, the EM auxiliary function is equivalent to the log-likelihood of a point-measurement filtering problem. This is achieved by factorising (12) and completing the square. Essentially (12) is a sum of quadratics in the target state which may be collected into a single quadratic, which is equivalent to the log of a normal distribution.

The same method can be used for a Gaussian measurement function where the mean is a non-linear function of the target state, i.e., $G(\tau \mid \mathbf{x}) = \mathcal{N}(\tau; \mathbf{h}(\mathbf{x}), \mathsf{R})$, with $\mathbf{h}(\mathbf{x})$ any function of the target state. In this case completing the square leads to an equivalent point-measurement problem where the measurements are non-linear and the noise is Gaussian [6]. The original non-linear relationship between the mean of the target point-spread-function and the target state is preserved as the non-linear point-measurement function, i.e.,

$$\sum_{l \in \mathcal{S}} \pi_t^{m(i)} \bar{z}_{tl} \int_{B_l} G^m(\tau \mid x_t^{m(i)}) \log\{G^m(\tau \mid x_t^m)\} d\tau \rightarrow \log\{\mathcal{N}(\tilde{\mathbf{z}}; \mathbf{h}(\mathbf{x}), \tilde{\mathsf{R}})\}. \tag{34}$$

Depending on the non-linearity, this might be solved by analytic linearisation (an extended Kalman Filter) or Monte Carlo methods.

For a target signature that is non-Gaussian, it becomes more difficult. For an arbitrary target, the integral in (34) is intractable. One solution is to use a numerical approximation: in [6] a grid-based approximation was used to evaluate the integral in (34) which was then used as an equivalent likelihood function to drive a particle filter. This method is capable of tracking any target structure, including a non-parametric shape estimate derived from data. However, it is numerically intensive. This section describes an alternative method that can used for non-Gaussian targets based on approximating the target signature as a Gaussian with spatially varying parameters.

5.1. Cell-Varying Point Spread Function

Unfortunately, the application that follows has a more complicated measurement function than above. In the application, the sensor builds up an image by sweeping an antenna across azimuth and measuring a range profile at each position. This creates an array of data in range bins and azimuths, but the data from different azimuths is collected sequentially, not simultaneously. This means that the airborne sensor platform

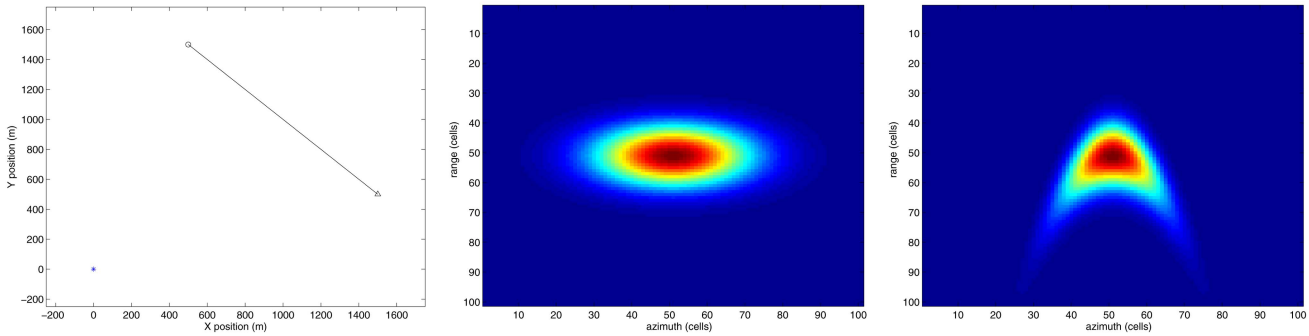


Fig. 6. Point spread function distortion due to sensor motion. (a) Target-sensor geometry. (b) Fixed sensor. (c) Moving sensor.

moves between azimuth collects, and the relative target position changes. The sensor response is a non-linear function of the relative target position. One way to model this is through a cell-varying target response: $G(\tau \mid x) \rightarrow G_l(\tau \mid x)$.

It can be shown (see Appendix A.1 for details) that the point-measurement result (34) can be further extended to the cell-varying psf situation. Provided that the target response can be expressed as a Gaussian function with a cell-dependent non-linear mean, $G_l(\tau \mid \mathbf{x}) = \mathcal{N}(\tau; \mathbf{h}_l(\mathbf{x}), \mathsf{R})$, then there exists an equivalent point-measurement problem with

$$\sum_{l \in \mathcal{S}} \pi_t^{m(i)} \bar{z}_{tl} \int_{B_l} G_l^m(\tau \mid x_t^{m(i)}) \log\{G_l^m(\tau \mid x_t^m)\} d\tau \rightarrow \log\{\mathcal{N}(\tilde{\mathbf{z}}_t^m; \tilde{\mathbf{h}}_t^m(\mathbf{x}_t^m), \tilde{\mathsf{R}}_t^m)\}. \quad (35)$$

Refer to $\tilde{\mathsf{R}}_t^m$, $\tilde{\mathbf{h}}_t^m(\mathbf{x})$ and $\tilde{\mathbf{z}}_t^m$ as the synthetic-measurement-covariance, synthetic-mean and synthetic-measurement respectively. For clarity, suppress the time and target indices.

The synthetic measurement covariance is the same as for the linear case, given in (15), namely

$$\tilde{\mathsf{R}} = \frac{1}{p_t^{m(i)}} \mathsf{R} \equiv (\pi_t^{m(i)} w)^{-1} \mathsf{R}$$

where $w = \sum_{l \in \mathcal{S}} w_l$, and $w_l = \bar{z}_{tl} P_l^m(x_t^{m(i)})$.

The synthetic mean is defined as

$$\tilde{\mathbf{h}}^{\mathrm{T}} \mathsf{R}^{-1} \tilde{\mathbf{h}} = \frac{1}{w} \sum_{l \in \mathcal{S}} w_l \mathbf{h}_l^{\mathrm{T}} \mathsf{R}^{-1} \mathbf{h}_l. \quad (36)$$

If the psf is separable, as in the previous section, then R will be diagonal, and (36) simplifies to

$$\tilde{\mathbf{h}}(j) = \sqrt{\frac{1}{w} \sum_{l \in \mathcal{S}} w_l \mathbf{h}_l(j)^2}. \quad (37)$$

That is, $\tilde{\mathbf{h}}$ is a weighted RMS average of the $\mathbf{h}_l$ values.

The synthetic measurement is defined as

$$\tilde{\mathbf{h}}^{\mathrm{T}} \mathsf{R}^{-1} \tilde{\mathbf{z}} = \frac{1}{w} \sum_{l \in \mathcal{S}} w_l \mathbf{h}_l^{\mathrm{T}} \mathsf{R}^{-1} \tilde{\mathbf{z}}_l \quad (38)$$

where $\tilde{\mathbf{z}}_l$ is the cell-level centroid for the target at cell l and is given by (14). Again, if the psf is separable, then R will be diagonal, and (38) simplifies to

$$\tilde{\mathbf{z}}(j) = \frac{1}{w \tilde{\mathbf{h}}(j)} \sum_{l \in \mathcal{S}} w_l \mathbf{h}_l(j) \tilde{\mathbf{z}}_l(j). \quad (39)$$

5.2. Application Example

For the application in this paper, the sensor forms range and azimuth cells and it moves as it collects the data. We now demonstrate how the cell-dependent psf model just described can be used to account for this motion.

Let (u_{tl}^s, v_{tl}^s) denote the Cartesian position of the sensor when cell l was observed. These locations change from one scan to the next and within the scan (hence the indices). The target state is in a Cartesian frame $\mathbf{x}_t^m = [u_t^m, \dot{u}_t^m, v_t^m, \dot{v}_t^m]^{\mathrm{T}}$ but the sensor response is polar, based on the range and bearing to the target when the particular cell was formed.

Note that both the target and the sensor move as a function of time but only the sensor motion has been modeled at a cell-to-cell level. The target has been implicitly assumed to be at the same position for every cell l. This is not correct because it's a moving target, but the target speed in the application is small and the effect of target motion will be assumed to be minor.

Because the sensor moves between collecting cells, the range and bearing equation is cell-dependent and is given by

$$\mathbf{h}_l(\mathbf{x}_t^m) = \begin{Bmatrix} \sqrt{(u_t^m - u_{tl}^s)^2 + (v_t^m - v_{tl}^s)^2} \\ \arctan\left(\dfrac{v_t^m - v_{tl}^s}{u_t^m - u_{tl}^s}\right) \end{Bmatrix}. \quad (40)$$

The effect of this sensor movement is therefore to distort the psf. Fig. 6 illustrates the difference between the psf for a fixed sensor and for a moving one. Fig. 6(a) shows the target-sensor geometry, (b) shows the psf for a fixed sensor at (1000, 1000) and (c) shows the moving sensor psf. This is an exaggerated example, but illustrates the potential effect of sensor motion.

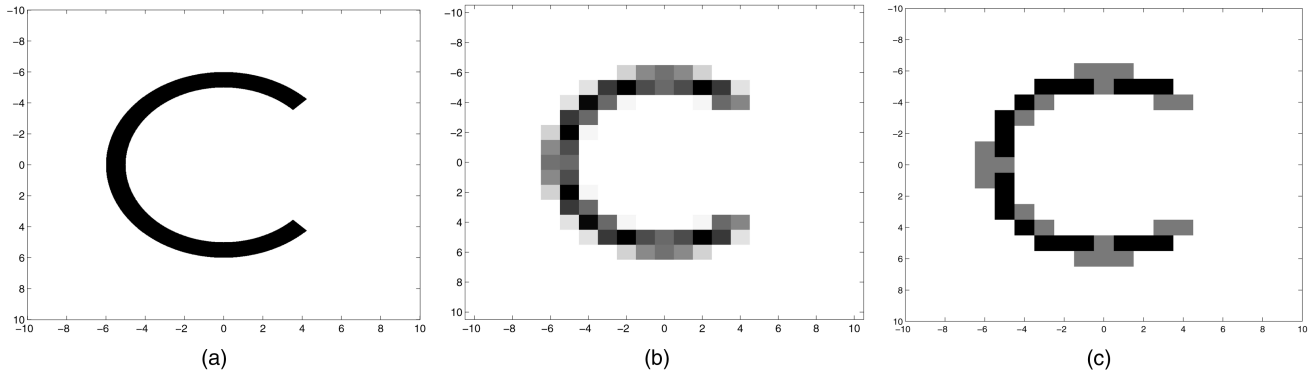


Fig. 7. Non-Gaussian point spread function. (a) Continuous-domain target psf. (b) Target response per cell. (c) Cell-varying spread function.

Substituting the cell-dependent $\mathbf{h}_l(\mathbf{x}_t^m)$ definition into the synthetic measurement leads to the equivalent measurement function

$$\tilde{\mathbf{h}}(\mathbf{x}) = \begin{Bmatrix} \left(\frac{1}{w}\sum_l w_l[(u_t^m - u_{tl}^s)^2 + (v_t^m - v_{tl}^s)^2]\right)^{1/2} \\ \left(\frac{1}{w}\sum_l w_l \left[\arctan\left(\frac{v_t^m - v_{tl}^s}{u_t^m - u_{tl}^s}\right)\right]^2\right)^{1/2} \end{Bmatrix} \tag{41}$$

which is the weighted root-mean-square range and bearing.

The estimation problem has been transformed from an image-measurement with target distortion due to motion, into a point-measurement with Gaussian noise. The implementation demonstrated in Section 6 used an extended Kalman Filter (EKF) to deal with the nonlinearity [3]. Although the weighting values, w_l, are data dependent, the synthetic measurement function may be constructed analytically, so it is possible to determine a Jacobian.

The sensor positions in (41) are known constants and the derivatives required for the Jacobian can be shown to be

$$\frac{\partial \tilde{\mathbf{h}}(\mathbf{x})[1]}{\partial u_t^m} = \frac{1}{\tilde{\mathbf{h}}(\mathbf{x})[1]}\left(u_t^m - \frac{1}{w}\sum_l w_l u_{tl}^s\right)$$

$$\frac{\partial \tilde{\mathbf{h}}(\mathbf{x})[1]}{\partial v_t^m} = \frac{1}{\tilde{\mathbf{h}}(\mathbf{x})[1]}\left(v_t^m - \frac{1}{w}\sum_l w_l v_{tl}^s\right)$$

$$\frac{\partial \tilde{\mathbf{h}}(\mathbf{x})[2]}{\partial u_t^m} = -\frac{1}{\tilde{\mathbf{h}}(\mathbf{x})[2]}\frac{1}{w}\sum_l w_l \frac{v_t^m - v_{tl}^s}{(u_t^m - u_{tl}^s)^2 + (v_t^m - v_{tl}^s)^2}$$

$$\frac{\partial \tilde{\mathbf{h}}(\mathbf{x})[2]}{\partial v_t^m} = \frac{1}{\tilde{\mathbf{h}}(\mathbf{x})[2]}\frac{1}{w}\sum_l w_l \frac{u_t^m - u_{tl}^s}{(u_t^m - u_{tl}^s)^2 + (v_t^m - v_{tl}^s)^2}$$

which are unsurprisingly very similar to the terms found in a standard range-bearing point-measurement problem [3].

5.3. Numerical Illustration

The result above may seem obscure and perhaps only useful in special situations. However, it can actually be used as a very general tool for approximating non-Gaussian targets. In [6] the particle filter was used to track a non-Gaussian target with H-PMHT. The target in that example was chosen look look like the letter C. It was shown that a non-Gaussian H-PMHT was far superior to the Gaussian approximation on several examples, including crossing targets and track initiation. We now demonstrate that this same letter C target can be tracked well using a cell-varying Gaussian point spread function. The purpose here is to qualitatively indicate the utility of the approach, not quantitatively assess it.

The letter-C target signature is given in radial coordinates by

$$h(r,\theta) = \begin{cases} A & \text{if} \quad 5 \geq r \geq 6 \quad \text{and} \quad |\theta| > \dfrac{\pi}{4} \\ 0 & \text{otherwise} \end{cases} \tag{42}$$

where A is a normalising constant. This response is shown in Fig. 7(a). The contribution of the target to each pixel is the integral of $h(r,\theta)$ over that pixel. An example of this is shown in Fig. 7(b).

Approximate the target signature by

$$h_l(x,y) = \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_l \\ y_l \end{bmatrix} \tag{43}$$

where x_l and y_l define a cell varying mean.

Denoting the coordinates of cell l as (i_l, j_l), and their radial equivalent coordinates as (r_l, θ_l), then an appropriate cell varying mean is

$$\begin{bmatrix} x_l \\ y_l \end{bmatrix} = \begin{cases} \begin{bmatrix} i_l \\ j_l \end{bmatrix} & \text{if} \quad 5 \geq r_l \geq 6 \\ & \text{and} \quad |\theta_l| > \dfrac{\pi}{4} \\ \mathsf{H}\mathbf{x}_t^m + 5.5\begin{bmatrix} \cos(\theta_l) \\ \sin(\theta_l) \end{bmatrix} & \text{if} \quad 0.9\times 5 \geq r_l \geq 1.1 \times 6 \\ & \text{and} \quad |\theta_l| > 0.9 \times \dfrac{\pi}{4} \\ \begin{bmatrix} -\infty \\ -\infty \end{bmatrix} & \text{otherwise} \end{cases} \tag{44}$$

with H the common cartesian position-only measurement matrix.

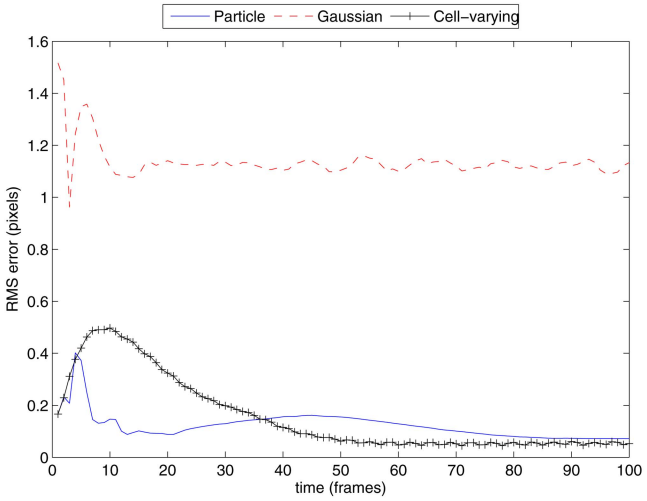


Fig. 8. Non-Gaussian target estimation accuracy.

Fig. 9. Small speedboat target.

The second tier allows for a graduated step down from the C rather than a sudden drop. This gives the algorithm an opportunity to correct for initialisation errors. A similar method was used in [6]. The resulting approximate target response is shown in Fig. 7(c).

The cell-varying approximation to the C psf was implemented and compared with the particle filter algorithm of [6]. Several scenarios were considered in [6], but for brevity only one of them is repeated here. This test scenario contains a single target under constant velocity motion, which the tracker must detect and track. The performance measure is the 2-D RMS estimation error averaged over 100 Monte Carlo trials. For this scenario there was a significant degradation in performance when the C psf was estimated with a single Gaussian compared with the particle filter H-PMHT implementation. Both of these RMS curves are shown in Fig. 8 along with the performance attained with the cell-varying approximation.

It is clear that the cell-varying approximation gives similar performance to the particle filter in this case. Although neither of the algorithm implementations has been optimised, it is informative to look at the computation cost. For 500 particles, the H-PMHT particle implementation took 430 seconds per monte carlo trial, compared with 3.4 seconds for the cell varying Gaussian and 1.7 seconds for the fixed Gaussian. Obviously this analytic approximation method has been able to achieve similar performance as numerical approximation at a fraction of the cost.

The results here show only a simple scenario where it was straightforward to devise a mapping of the non-Gaussian target psf to a cell-varying Gaussian one. In a more realistic situation it may be a more difficult task. Nevertheless, the example illustrates the power of this method: for only a minor increase in computation cost it allows the use of an arbitrary psf, not merely a Gaussian one.

6. DETECTING A SMALL BOAT

An implementation of H-PMHT for detecting a boat in experimental data is now described. The data used was collected by DRDC in Halifax harbour, Canada, using the DRDC Ottawa X-band Wideband Experimental Airborne Radar. The sensor observed a small speedboat, as shown in Fig. 9. A detailed description of the experiment and the data characteristics can be found in [13], [11], [12].

Some salient features of the experimental data will now be discussed and a conventional sequential detect-then-track processing method described. The conventional algorithm is then compared with the H-PMHT output.

6.1. Data Characteristics

Due to the high sensor bandwidth, the range resolution of the sensor was quite fine compared with the physical extent of the target, and the resulting range profile was spread over many range bins. The sensor swept through azimuth and collected range profiles. Again, the azimuth resolution was relatively fine compared with the beam width of the sensor, so the target response was spread over many azimuth bins.

The volume of data from the sensor is relatively large: each scan consists of around 13000 range bins and over 400 azimuth bins. This makes it challenging to process the data in a timely manner and provides an excellent measure of the scalability of the H-PMHT approach to realistic surveillance volumes.

Fig. 10 shows a segment of one scan. Two strong returns are evident near the centre of the image. The rightmost return is from a buoy which happened to be near the target, and the leftmost one is the speedboat. Both buoy and boat have a response that is spread over tens of range bins and tens of azimuths. Numerous other vertical streaks are evident. These are due to sea clutter and are unfortunately target-like. Dividing the environment into background noise and interference spikes, it is clear that the signal to noise ratio is high.

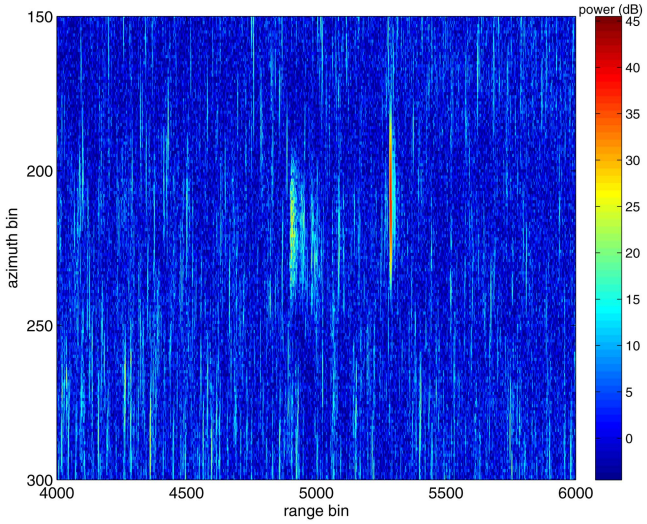


Fig. 10. Segment of a single scan showing target and clutter.

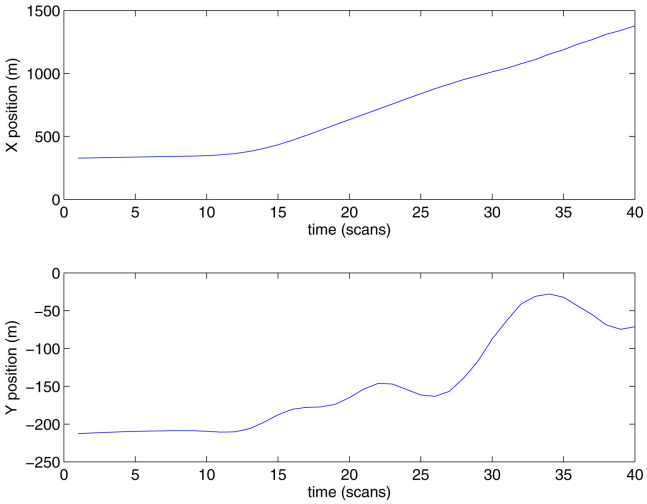


Fig. 11. Target trajectory (local cartesian frame).

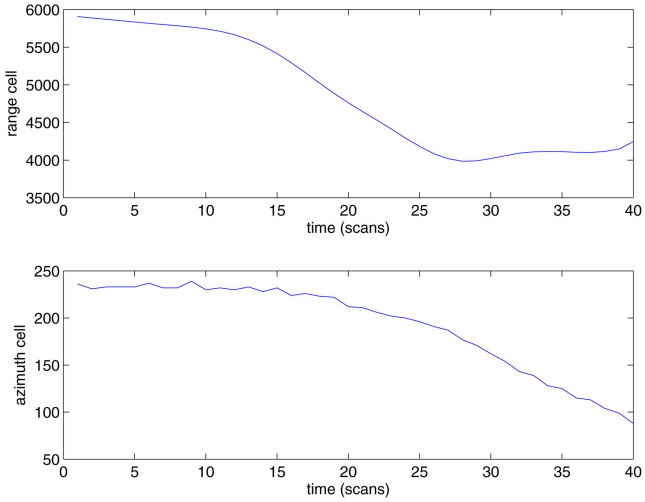


Fig. 12. Target trajectory (sensor frame).

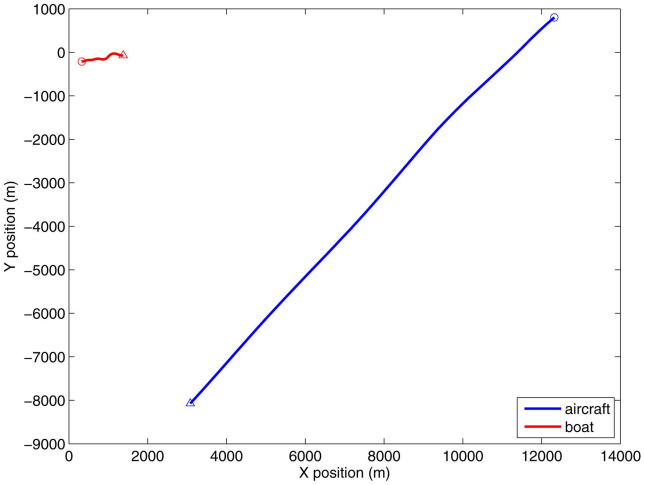


Fig. 13. Target-sensor geometry.

However, the signal to interference ratio is not. In fact the brightest spots on each image are usually sea clutter spikes.

McDonald has characterised the clutter distribution [13], [11], [12]. It was found to be well described by a KA distribution, which has significantly higher tails than the Rayleigh distribution typically assumed by radar applications of TkBD. This is of no direct consequence for H-PMHT since it does not use the data likelihood ratio, but it may be expected to lead to higher false track rates.

Fig. 10 also highlights that the target response in range is not simply a broad point spread function, but rather shows a much more complicated structure due to scattering off various parts of the boat. The target has the potential to be perceived by the sensor as several targets.

The speedboat was fitted with a GPS logger. However, registration of the GPS data with the radar data has proven difficult [13]. Thus the GPS data is useful as an indication of where the target is, but it is not useful from the perspective of determining target localisation error.

The data of interest is a sequence of 40 frames collected while the speedboat was manoeuvring. The boat was initially almost stationary for around 10 frames and then it accelerated, following a snaking trajectory for the remaining frames. The signal to noise ratio during the first part of the trajectory was relatively poor, but increased when the speedboat moves more quickly.

Fig. 11 shows the speedboat trajectory in local cartesian co-ordinates and Fig. 12 shows the trajectory in the measurement frame (range and azimuth cells). These figures highlight the volatility of the target motion and also demonstrate that the target was moving very rapidly through the sensor field of view. Between scans 10 and 25, the target moves through roughly 1500 range cells.

Fig. 13 shows the overall scenario geometry in local cartesian co-ordinates. Circles mark the starting position of the aircraft and the boat and triangles mark their final position. The aircraft maintained an approximately constant altitude of 300m and moved in an approximately straight path.

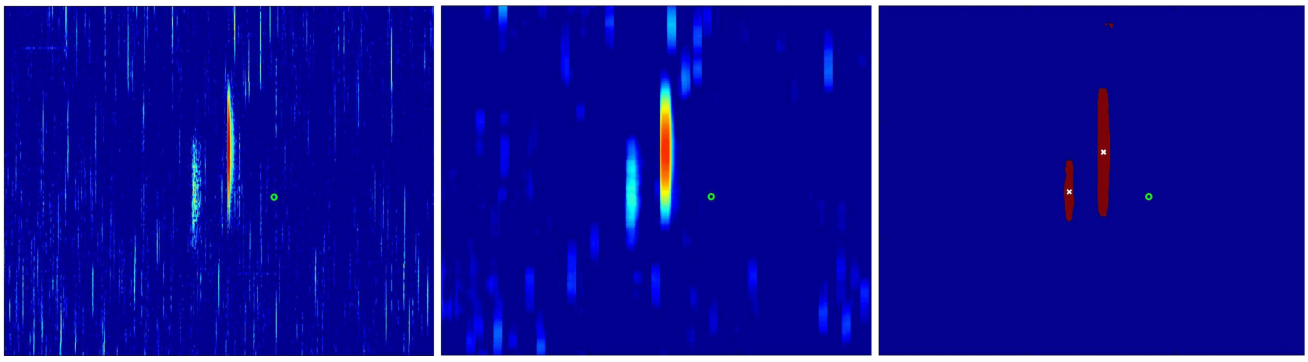


Fig. 14. Target trajectory (sensor frame). (a) Input sensor image. (b) Smoothed sensor image. (c) Segmentation mask.

6.2. Conventional Detection and Tracking

The conventional approach is to apply a single frame detector followed by a point-measurement tracking algorithm. In this case, the size of the target response is relatively large compared with the sensor resolution, so it is appropriate to use an image segmentation based detector. The general strategy is to find the centroids of extended objects and track them using a Probabilistic Data Association Filter (PDAF), similar to [15].

A significant effort was spent in tuning the preprocessing before applying the PDAF. First, it was observed that the target signal was spread over a greater number of image cells than the impulsive noise returns. This motivated the use of spatial averaging, which was implemented by convolving the sensor image with a 10 beam by 5 range uniform rectangular kernel. A threshold was then applied to the smoothed image and adjacent pixels that exceed the threshold using a two-pass labeling algorithm [9] (this can be efficiently implemented in Matlab via the *bwlabel* function). The centroid of each connected set of pixels was determined and this centroid was mapped to a cartesian reference space. Measurements were only retained if the number of foreground pixels (i.e., threshold crossings) associated with a particular object was within an upper and lower tolerance band. This single-frame detection scheme was very CPU-intensive to perform, but yielded a high probability of detection for very few false alarms. From this point onwards, it was straightforward to form tracks, so only a very simple point-tracker was considered.

Fig. 14 illustrates the preprocessing used to feed the conventional tracker. Fig. 14(a) shows the raw radar image with a green circle at the GPS location, which is close to the target but not co-located with it. Fig. 14(b) shows the result of the smoothing filter and (c) shows the output of the segmentation algorithm. The larger target near the middle of the image is the buoy and the smaller one to the left of the buoy is the boat.

Each measurement was transformed into a local cartesian reference frame on the ground using the known sensor location. The tracker assumed a cartesian measurement model with a fixed sensor variance, R, as given in Table I. The X-direction in the local frame was approximately aligned with the radial between the target and sensor (although this varied with time) and so the measurement noise was smaller in the X-direction than the Y-direction.

The PDAF used was a simplified version of the algorithm described in [5] and differed from a text-book algorithm (e.g. [1]) only in that it used target *visibility* to determine the merit of tracks for track maintenance decisions. Uniform clutter was assumed, the target state was Cartesian position and velocity in two dimensions and an almost-constant-velocity model was used. The process noise for this model, Q, is given in Table I.

Thus the measurement association weights were given by

$$\beta_i = \frac{b_i}{\sum_{i=-1}^{n_t} b_i} \tag{45}$$

where n_t is the number of measurements in the scan and

$$b_i = \begin{cases} \dfrac{1 - Pv_{t|t-1}}{Pv_{t|t-1}} & i = -1 \\ 1 - Pd & i = 0 \\ \dfrac{PdV}{|2\pi\mathsf{S}|}\exp\{-\frac{1}{2}d_i^2\} & 0 < i \le n_t \end{cases} \tag{46}$$

with Pd the probability of detection, V the surveillance volume, S the innovation covariance, $d_i^2 = (\mathsf{H}\mathbf{x} - \mathbf{z}_i)^{\mathrm{T}}\mathsf{S}^{-1}(\mathsf{H}\mathbf{x} - \mathbf{z}_i)$ the statistical distance between the track and measurement i, and $Pv_{t|t-1}$ the predicted target visibility, given by

$$Pv_{t|t-1} = (1 - P_{\text{death}})Pv_{t-1|t-1} + P_{\text{rebirth}}(1 - Pv_{t-1|t-1}) \tag{47}$$

where the probability of target death and rebirth are tuning parameters given in Table I. The updated target visibility is given by $Pv_{t|t} = 1 - \beta_{-1}$ and is used as the basis of track management decisions. Similar to the H-PMHT track manager, candidate tracks are formed using 2-point differencing.

Table I lists the parameters of the conventional detection and tracking approach.

TABLE I
Conventional Detection and Tracking Parameters

Detector Parameters	
minimum segment size	75 pixels
maximum segment size	none
segmentation threshold	$2 \times$ image mean
Tracker Parameters	
track state vector	$[x, \dot{x}, y, \dot{y}]^{\mathrm{T}}$ x and y in metres $\dot{x}$ and $\dot{y}$ in metres per frame
measurement vector	$[x, y]^{\mathrm{T}}$ in metres
R	$\begin{bmatrix} 400 & 0 \\ 0 & 900 \end{bmatrix}$
Q	$100 \begin{bmatrix} \mathrm{Q2} & 0 \\ 0 & \mathrm{Q2} \end{bmatrix}$
Q2	$\begin{bmatrix} \frac{1}{3}\Delta_t^3 & \frac{1}{2}\Delta_t^2 \\ \frac{1}{2}\Delta_t^2 & \Delta_t \end{bmatrix}$
P_{death}	0.012
P_{rebirth}	0
initial visibility	$Pv_0 = 0.5$
promotion threshold	$Pv_{t\|t} > 0.6$ for any $t > 4$ frames

TABLE II
H-PMHT Parameters

Detector Parameters	(used to form candidate tracks)
minimum segment size	75 pixels
maximum segment size	none
segmentation threshold	$3 \times$ image mean
H-PMHT(rb) Parameters	
track state vector	$[r, \dot{r}, b, \dot{b}]^{\mathrm{T}}$ r and b in cells $\dot{r}$ and $\dot{b}$ in cells per frame
Q	$\begin{bmatrix} 100\mathrm{Q2} & 0 \\ 0 & 10^4\mathrm{Q2} \end{bmatrix}$
H-PMHT(xy) Parameters	
track state vector	$[x, \dot{x}, y, \dot{y}]^{\mathrm{T}}$ x and y in metres $\dot{x}$ and $\dot{y}$ in metres per frame
Q	$1.2 \times 10^5 \begin{bmatrix} \mathrm{Q2} & 0 \\ 0 & \mathrm{Q2} \end{bmatrix}$
Common Parameters	
measurement vector	$[r, b]^{\mathrm{T}}$ in cells
R	$\begin{bmatrix} 400 & 0 \\ 0 & 900 \end{bmatrix}$
confirmation threshold	4 frames greater than 20 dB

6.3. Specific H-PMHT models and Parameters

The H-PMHT was implemented as a time-recursive filter for this analysis, that is, there was no batch; each scan was processed sequentially and only one scan was available to the algorithm at a time. The state estimates were not smoothed.

Two processing strategies for H-PMHT were considered. In the first, the movement of the sensor was ignored, and the target state was modeled in the measurement frame. That is, the state was pixels and pixels per frame. This makes life much easier for implementation, and the separable point spread function expressions may be used. This version is referred to as H-PMHT(rb) since the target state is in range-beam space.

The second strategy was to model the target state in Cartesian coordinates on the ground and use the cell-dependent non-linear method to relate the target state to the sensor image. This approach has much higher implementation complexity but should more accurately model the true system. This version is referred to as H-PMHT(xy).

The H-PMHT(rb) software was a Matlab implementation of the equations derived in Section 3.4. There are two integrals that need to be implemented and these are vector versions of the per-cell contribution of each target in (6)

$$P_l^m(\mathbf{x}_t^m) = \int_{B_l} G^m(\tau \mid \mathbf{x}_t^m) d\tau$$

and the cell-level centroid in (14)

$$\tilde{z}_{tl}^m = \frac{1}{P_l^m(x_t^m)} \int_{B_l} \tau G^m(\tau \mid x_t^m) d\tau.$$

Appendix A.2 provides some discussion on implementing these integrals. Besides those two expressions, the remaining software is a direct encoding of Section 2.4.

The background was assumed to be uniform and the target was assumed to be a simple Gaussian with a diagonal covariance matrix as given in Table II. The target was more spread in azimuth than range, so the azimuth variance was higher.

The target model was an almost constant velocity model in the plane. That is, the target motion was approximated with a constant rate of movement through range and azimuth bins. The target state was in units of bins for position and bins-per-scan for velocity. This was transformed into ground units as a post-processing stage for comparison with the other trackers and the ground truth. The process noise variance used is given in Table II, also using bins and scans as units.

The efficiency measures described in Section 3 were employed, including gating and a per-target convergence test. A maximum of 10 EM iterations was performed per frame. The single frame detector used for the PDA was also used here to seed candidate tracks which were initialised using two-point differences.

Candidates were promoted if they had an estimated SNR of greater than 20 dB for four frames and terminated if they had an estimated SNR of less than -5 dB for two consecutive frames.

The H-PMHT(xy) software was a Matlab implementation of the cell-varying psf equations in Section 5.2. This algorithm used the same measurement covariance

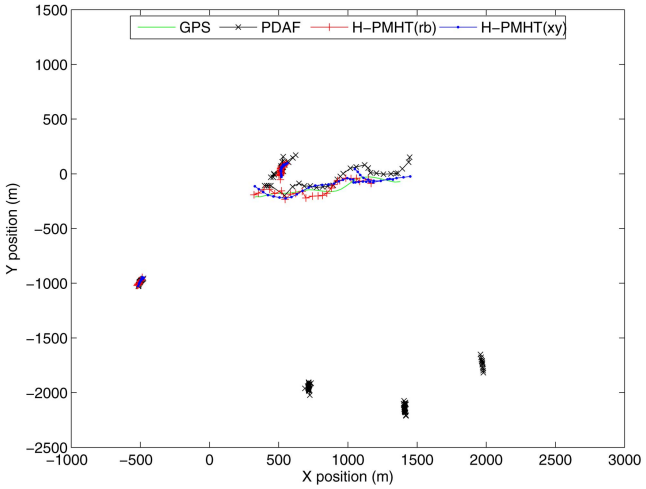


Fig. 15. Small speedboat target.

matrix as the measurement-space implementation and also used an almost constant velocity target model. However, the target state was in different units, namely metres, so a different process noise covariance matrix was required, as given in Table II. The same track management rules were used as for the measurement-space implementation.

For the H-PMHT(xy), the position part of the state vector was in metres, but the measurement frame was in range and azimuth cells. The estimator was therefore implemented as an EKF as described in Section 5.2. The measurement model included the scaling and offset from range in metres to range in cells and similarly for azimuth. The per-cell contribution of the target P_l^m was calculated as described in Appendix A.2, but the cell-centroids were replaced by the cell-dependent terms in Section 5.2.

6.4. Results

The tracking outputs of the PDAF based approach and the two H-PMHT approaches are now presented. Each are compared with GPS data collected from the target of interest.

Fig. 15 shows all of the output tracks from each of the three trackers overlaid with the GPS measurements. The target tracks are roughly centred in the plot and move from left to right. As mentioned earlier, there was a buoy that was coincidentally in the region, near to the target. The buoy location is slightly north of the target starting position. There are no other known targets in the area, and the other tracks are assumed to be false.

All three trackers were able to detect the speedboat and the buoy. The PDAF tracker shows a number of false tracks to the south of the area, which is closer in range to the sensor. At close range, the clutter spikes are more prominent and so the tracker forms false tracks.

By manual inspection, the tracks that have followed the target were determined and are shown on Fig. 16. It is clear that none of the trackers were able to detect the target during the initial period where it was stationary.

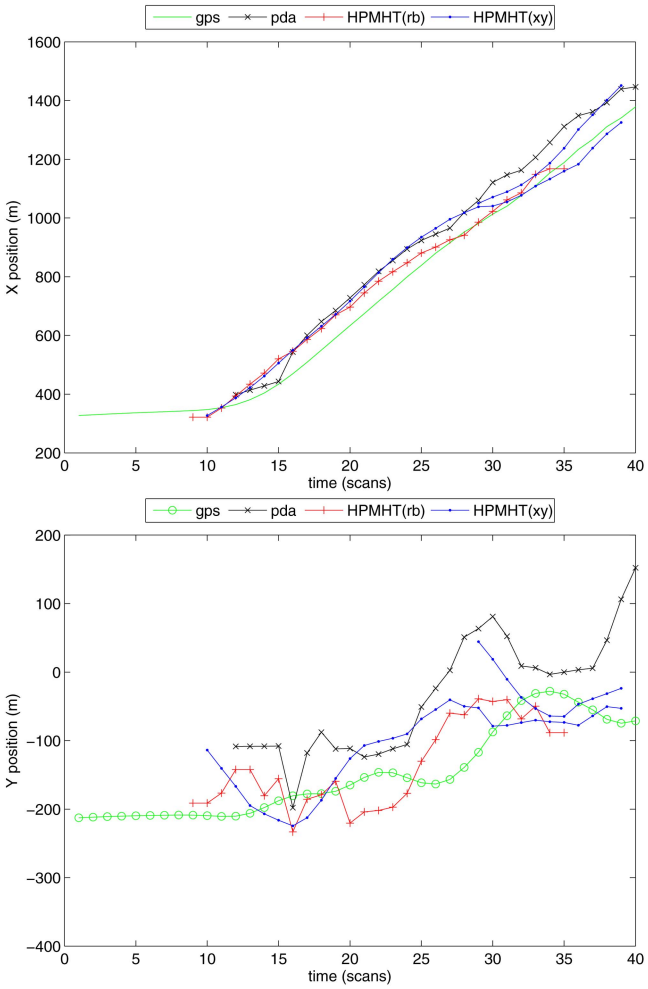


Fig. 16. Small speedboat target.

During this time it is not possible to detect the target by eye in the data either: the received SNR is very low. Once it starts moving and the SNR improves, the PDAF tracker took a little longer to establish track.

At approximately scan 32, the target performed a manoeuvre that caused the PDAF tracker to diverge. Unfortunately, soon after this the target left the sensor coverage area in scan 39 and there is insufficient data to determine whether the PDAF would have recovered from the error. The measurement-space H-PMHT(rb) tracker lost the target at the same time. The ground-space H-PMHT(xy) followed the target until the second to last scan. However during the last ten scans it produced a duplicate track on the target. The original track started out following the centre of mass of the target response, but later it drifted slightly to focus on the short-range component (recall that the target response is a complicated superposition of multiple partially resolved scatterers). This caused the tracker to form a second track on the longer-range component of the target response.

Fig. 17 shows tracks that were manually determined to be following the buoy. The buoy has an extremely high SNR and is easily tracked by all of the algorithms.

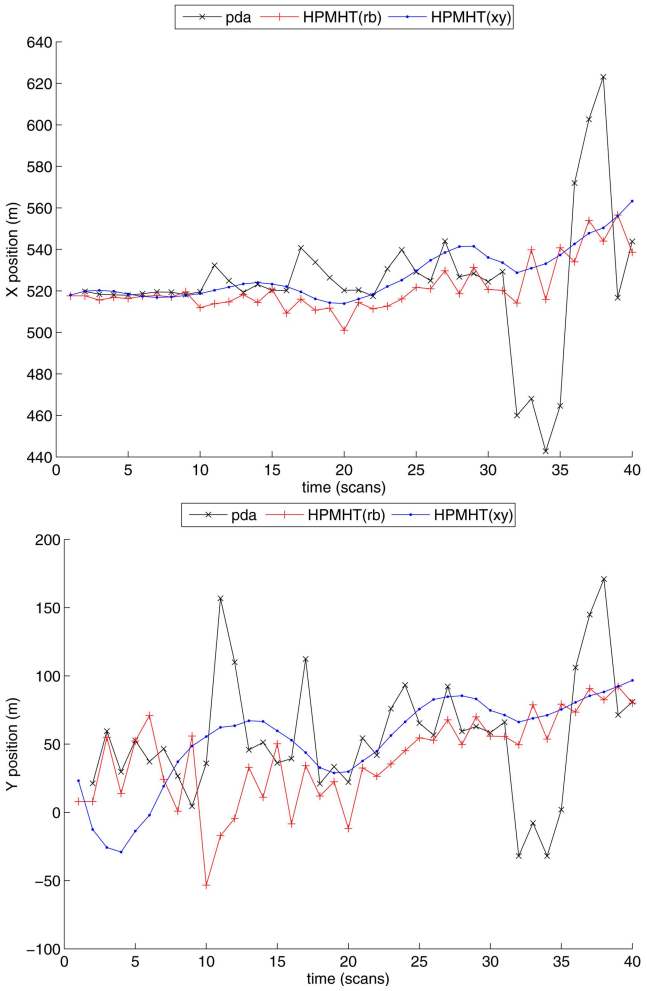


Fig. 17. Strong buoy target of opportunity.

At times, the peak level of the buoy is 40 dB higher than the boat. However, the H-PMHT(xy) track is considerably smoother than the others, especially the PDAF which shows several large excursions. This is partly because the PDAF was tuned with a relatively high process noise to cope with the high manoeuvrability of the speedboat and partly because the detection process reports the centroid of a group of connected pixels, which can be highly variable.

The implementation of the algorithms was not optimised, so the CPU time spent by each is not a reliable measure of performance. Nevertheless, figures are given as a qualitative comparison. The H-PMHT(rb) was the fastest algorithm, requiring 280 CPU seconds to run under Matlab on a quad core PC. For the PDAF approach, the single-frame detector was orders of magnitude more expensive than the PDAF tracker. This is because of the very large number of pixels in the sensor image and the double-pass clustering required to extract detections. In fact, the single-frame detector took around twice the computation effort of the measurement-space H-PMHT. Substantial experimentation showed that simpler detectors for the PDAF lead to an abundance of false tracks. The PDAF total cost was 478 CPU seconds.

The ground-space H-PMHT(xy) was the slowest at 575 CPU seconds. Since the sensor image is so large, the available memory limited the number of intermediate variables that could be stored. In particular, for candidate tracks, it was not feasible to store the P_l^m values, which meant that they had to be calculated twice for each EM iteration. These values alone can amount to around half a gigabyte of storage for a modest number of tracks. Thus the trade-off between memory and computation resulted in a longer execution time for the ground-space H-PMHT. For the sensor-space H-PMHT, these quantities can be stored as their marginal vectors, which are a fraction of a percent of the size. An optimised implementation of the ground-space algorithm could alleviate much of this overhead.

7. SUMMARY

This article has reviewed the Histogram PMHT algorithm from an implementation perspective. It addresses some of the practical issues arising from real applications: efficient implementation, track management, and non-linearity. The purpose has been to demonstrate that the encouraging simulation results obtained in recent comparisons on thumbnail images with benevolent targets do translate to more realistic conditions.

The H-PMHT algorithm was applied to an experimental data set collected by DRDC containing a manoeuvring speedboat amongst highly challenging sea clutter and close to a strong secondary target: a buoy. The buoy return was at times 40 dB higher than the speedboat, which had a peak signal-to-interference ratio of less than 0 dB. Nevertheless, the H-PMHT algorithm was able to detect the boat and follow its manoeuvres while forming almost no false tracks. The H-PMHT demonstrated a good ability to track multiple closely spaced targets with a high dynamic range.

The experimental data was also processed via a more conventional method, using image segmentation and a PDAF. The conventional output took marginally longer to establish track on the target and produced more false tracks. The conventional detector was very computationally demanding so the overall cost of the H-PMHT was less than the conventional approach.

The good performance of H-PMHT illustrates that the algorithm is not sensitive to the assumed target model, since the speedboat target response was a complex interaction of multiple scatterers rather than a simple Gaussian. It was also not sensitive to target-like clutter responses in the experimental data. Although the H-PMHT does not explicitly model the clutter amplitude likelihood, intuition suggests that the maths must make some implicit modeling assumption, which would doubtlessly be broken by this data. Initialisation was also poor since the target moved very quickly through the image at an average of one hundred range bins per scan.

In addition to the experimental case study, several important results were introduced. An efficient matrix representation of the algorithm was derived for two-dimensional sensor images and when combined with appropriate gating was demonstrated to give an algorithm independent of sensor image size for a fixed resolution. Track management was discussed in detail, a function that recent papers describe as lacking from H-PMHT. Finally a novel measurement model was introduced as a method for dealing with non-Gaussian targets by transforming them into a Gaussian target with a spatially varying mean. This model was demonstrated on a simple simulation and used to compensate for ownship motion in the experimental data.

This article has demonstrated that H-PMHT is an effective algorithm for automatically extracting multiple targets against a very challenging background and can process very high data volumes at a reasonable rate.

8. ACKNOWLEDGMENT

The author would like to thank Defence Research and Development Canada for providing the test data set and in particular Dr Michael McDonald for his assistance in interpreting and processing the data.

APENDIX

A.1. Cell-varying Point Spread Function Derivation

Consider the measurement component of the target auxiliary function (12)

$$\sum_{t=1}^{T}\sum_{l\in\mathcal{S}}\pi_t^{m(i)}\bar{z}_{tl}\int_{B_l} G_l^m(\tau\mid\mathbf{x}_t^{m(i)})\log\{G_l^m(\tau\mid\mathbf{x}_t^m)\}d\tau. \tag{48}$$

For clarity, consider one term from the temporal summation and suppress the target and time indices, m and t. Abbreviate $G_l^{(i)}\equiv G_l^m(\tau\mid\mathbf{x}_t^{m(i)})$. Using the cell-dependent Gaussian model, one time-slice of (48) becomes

$$\begin{aligned}
&\pi^{(i)}\sum_{l\in\mathcal{S}}\bar{z}_l\int_{B_l} G_l^{(i)}\log\{\mathcal{N}(\tau;\mathbf{h}_l,\mathsf{R})\}d\tau\\
&\quad=-\frac{1}{2}\pi^{(i)}\sum_{l\in\mathcal{S}}\bar{z}_l\int_{B_l} G_l^{(i)}(\tau-\mathbf{h}_l)^{\mathrm{T}}\mathsf{R}^{-1}(\tau-\mathbf{h}_l)d\tau+C\\
&\quad=-\frac{1}{2}\pi^{(i)}\sum_{l\in\mathcal{S}}\bar{z}_l\left\{\int_{B_l} G_l^{(i)}\mathbf{h}_l^{\mathrm{T}}\mathsf{R}^{-1}\mathbf{h}_l d\tau\right.\\
&\qquad\qquad\left.-2\int_{B_l} G_l^{(i)}\mathbf{h}_l^{\mathrm{T}}\mathsf{R}^{-1}\tau d\tau\right\}+C\\
&\quad=-\frac{1}{2}\pi^{(i)}\sum_{l\in\mathcal{S}}\bar{z}_l\left\{\mathbf{h}_l^{\mathrm{T}}\mathsf{R}^{-1}\mathbf{h}_l\int_{B_l} G_l^{(i)}d\tau\right.\\
&\qquad\qquad\left.-2\mathbf{h}_l^{\mathrm{T}}\mathsf{R}^{-1}\int_{B_l} G_l^{(i)}\tau d\tau\right\}+C\\
&\quad=-\frac{1}{2}\pi^{(i)}\left\{\sum_{l\in\mathcal{S}}\bar{z}_l P_l^m(\mathbf{x}^{(i)})\mathbf{h}_l(\mathbf{x})^{\mathrm{T}}\mathsf{R}^{-1}\mathbf{h}_l(\mathbf{x})\right.\\
&\qquad\qquad\left.-2\sum_{l\in\mathcal{S}}\bar{z}_l P_l^m(\mathbf{x}^{(i)})\mathbf{h}_l(\mathbf{x})^{\mathrm{T}}\mathsf{R}^{-1}\tilde{\mathbf{z}}_l\right\}+C
\end{aligned} \tag{49}$$

where C soaks up the terms that are constant with respect to $\mathbf{x}$, and

$$\tilde{\mathbf{z}}_l=\frac{1}{P_l^m(\mathbf{x}^{(i)})}\int_{B_l}\mathcal{N}(\tau;\mathbf{h}_l(\mathbf{x}^{(i)}),\mathsf{R})\tau d\tau \tag{50}$$

that is, the cell-level centroid defined in (14).

Let

$$w=\sum_{l\in\mathcal{S}}w_l=\sum_{l\in\mathcal{S}}\bar{z}_l P_l^m(\mathbf{x}^{(i)}). \tag{51}$$

The expression in (49) can be represented as a single quadratic by completing the square, i.e.,

$$\begin{aligned}
&-\frac{1}{2}\pi^{(i)}\left\{\sum_{l\in\mathcal{S}}w_l\mathbf{h}_l(\mathbf{x})^{\mathrm{T}}\mathsf{R}^{-1}\mathbf{h}_l(\mathbf{x})-2\sum_{l\in\mathcal{S}}w_l\mathbf{h}_l(\mathbf{x})^{\mathrm{T}}\mathsf{R}^{-1}\tilde{\mathbf{z}}_l\right\}+C\\
&\quad=-\frac{1}{2}(\tilde{\mathbf{z}}-\tilde{\mathbf{h}}(\mathbf{x}))^{\mathrm{T}}\tilde{\mathsf{R}}^{-1}(\tilde{\mathbf{z}}-\tilde{\mathbf{h}}(\mathbf{x}))+C'
\end{aligned} \tag{52}$$

where C' is another constant and the terms $\tilde{\mathbf{z}}$, $\tilde{\mathbf{h}}(\mathbf{x})$, and $\tilde{\mathsf{R}}$ are to be determined. These terms allow the measurement component of the auxiliary function to be expressed as an equivalent point-measurement.

Choose

$$\tilde{\mathsf{R}}^{-1}=\pi^{(i)}\sum_{l\in S}w_l\mathsf{R}^{-1}=\pi^{(i)}w\mathsf{R}^{-1}. \tag{53}$$

Equating the quadratic terms gives

$$\begin{aligned}
-\frac{1}{2}\tilde{\mathbf{h}}(\mathbf{x})^{\mathrm{T}}\tilde{\mathsf{R}}^{-1}\tilde{\mathbf{h}}(\mathbf{x})&=-\frac{1}{2}\pi^{(i)}\sum_{l\in S}w_l\mathbf{h}_l(\mathbf{x})^{\mathrm{T}}\mathsf{R}^{-1}\mathbf{h}_l(\mathbf{x})\\
\pi^{(i)}w\tilde{\mathbf{h}}(\mathbf{x})^{\mathrm{T}}\mathsf{R}^{-1}\tilde{\mathbf{h}}(\mathbf{x})&=\pi^{(i)}\sum_{l\in S}w_l\mathbf{h}_l(\mathbf{x})^{\mathrm{T}}\mathsf{R}^{-1}\mathbf{h}_l(\mathbf{x})\\
\tilde{\mathbf{h}}(\mathbf{x})^{\mathrm{T}}\mathsf{R}^{-1}\tilde{\mathbf{h}}(\mathbf{x})&=\frac{1}{w}\sum_{l\in S}w_l\mathbf{h}_l(\mathbf{x})^{\mathrm{T}}\mathsf{R}^{-1}\mathbf{h}_l(\mathbf{x})
\end{aligned} \tag{54}$$

which can be simplified to

$$\tilde{\mathbf{h}}(\mathbf{x})[j]=\sqrt{\frac{1}{w}\sum_{l\in S}w_l\mathbf{h}_l(\mathbf{x})[j]^2} \tag{55}$$

if R is diagonal. Here, $\mathbf{h}_l(\mathbf{x})[j]$ is the jth element of vector $\mathbf{h}_l(\mathbf{x})$.

Similarly, equating the linear terms

$$\begin{aligned}
\tilde{\mathbf{h}}(\mathbf{x})^{\mathrm{T}}\tilde{\mathsf{R}}^{-1}\tilde{\mathbf{z}}&=\pi^{(i)}\sum_{l\in S}w_l\mathbf{h}_l(\mathbf{x})^{\mathrm{T}}\mathsf{R}^{-1}\tilde{z}_l\\
\tilde{\mathbf{h}}(\mathbf{x})^{\mathrm{T}}\mathsf{R}^{-1}\tilde{\mathbf{z}}&=\frac{1}{w}\sum_{l\in S}w_l\mathbf{h}_l(\mathbf{x})^{\mathrm{T}}\mathsf{R}^{-1}\tilde{z}_l
\end{aligned} \tag{56}$$

which simplifies to

$$\tilde{\mathbf{z}}[j]=\frac{1}{w\tilde{\mathbf{h}}(\mathbf{x})[j]}\sum_{l\in S}w_l\mathbf{h}_l(\mathbf{x})^{\mathrm{T}}[j]\tilde{z}_l[j] \tag{57}$$

if R is diagonal.

A.2. H-PMHT integral simplifications

Implementation of H-PMHT requires the encoding of two integrals, namely the per-cell contribution of each target in (6)

$$P_l^m(\mathbf{x}_t^m) = \int_{B_l} G^m(\tau \mid \mathbf{x}_t^m) d\tau$$

and the cell-level centroid in (14)

$$\tilde{z}_{tl}^m = \frac{1}{P_l^m(\mathbf{x}_t^m)} \int_{B_l} \tau G^m(\tau \mid \mathbf{x}_t^m) d\tau.$$

Here analytic expressions are developed for each of these suitable for software implementation. It will be assumed that the sensor cells, B_l, conform to a regular grid and that the target psf is Gaussian, i.e.,

$$G^m(\tau \mid \mathbf{x}_t^m) = \frac{1}{|2\pi \mathsf{R}|} \exp\left\{ -\frac{1}{2} (\mathsf{H}\mathbf{x}_t^m - \tau)^{\mathrm{T}} \mathsf{R}^{-1} (\mathsf{H}\mathbf{x}_t^m - \tau) \right\}.$$

The integral in (6) is simply the area under the pdf for cell l and can be evaluated using erf functions since $G^m(\cdot)$ is Gaussian. However, these are relatively costly to evaluate. For efficiency, the implementation in this paper makes the approximation that $G^m(\cdot)$ is constant over B_l and so the approximate per-cell contribution is

$$P_l^m(\mathbf{x}_t^m) \approx \frac{\Delta_\tau}{|2\pi \mathsf{R}|} \exp\left\{ -\frac{1}{2} (\mathsf{H}\mathbf{x}_t^m - \bar{\tau}_l)^{\mathrm{T}} \mathsf{R}^{-1} (\mathsf{H}\mathbf{x}_t^m - \bar{\tau}_l) \right\}$$

with $\bar{\tau}_l$ the centre of cell l and Δ_τ the cell area, which is the same for all cells because of the regular grid. This approximation is reasonable when R is large compared to the cell size and the psf is thus slowly varying, as is the case for the application considered in this paper.

The cell-level centroid is the mean of the target psf over the cell. An expression is now developed for this for a one-dimensional psf. The two-dimensional version is built from two one-dimensional terms as described in Section 3. For simplicity of notation, assume that the target state is simply position, so H is unity. The extension to constant velocity is simply a matter of book-keeping. Since the psf is one-dimensional, R is a scalar.

Since $G^m(\cdot)$ is Gaussian it follows that

$$G^m(\tau \mid \mathbf{x}_t^m)' = \frac{dG^m(\tau \mid \mathbf{x}_t^m)}{d\tau} = G^m(\tau \mid \mathbf{x}_t^m) \left(\frac{\mathbf{x}_t^m}{\mathsf{R}} - \frac{\tau}{\mathsf{R}} \right)$$

which implies

$$\tau G^m(\tau \mid \mathbf{x}_t^m) = \mathbf{x}_t^m G^m(\tau \mid \mathbf{x}_t^m) - \mathsf{R} G^m(\tau \mid \mathbf{x}_t^m)'.$$

Substituting this back into the centroid definition gives

$$\begin{aligned} \tilde{z}_{tl}^m &= \frac{\mathbf{x}_t^m}{P_l^m} \int_{B^l} G^m(\tau \mid \mathbf{x}_t^m) d\tau - \frac{\mathsf{R}}{P_l^m} \int_{B^l} G^m(\tau \mid \mathbf{x}_t^m)' d\tau, \\ &= \mathbf{x}_t^m - \frac{\mathsf{R}}{P_l^m} \left[G^m\left(\bar{\tau}_l + \frac{\Delta_\tau}{2} \,\middle|\, \mathbf{x}_t^m \right) - G^m\left(\bar{\tau}_l - \frac{\Delta_\tau}{2} \,\middle|\, \mathbf{x}_t^m \right) \right] \end{aligned}$$

which is easily implemented in software.

REFERENCES

[1] Y. Bar-Shalom and X. R. Li
Multitarget-multisensor tracking: principles and techniques.
Storrs, CT: YBS, 1995.

[2] Y. Barniv
Dynamic programming algorithm for detecting dim moving targets.
In Y. Bar-Shalom, Ed., *Multitarget-Multisensor Tracking: Advanced Applications*, Artech House, 1990, ch. 4.

[3] S. S. Blackman and R. Popoli
Design and Analysis of Modern Tracking Systems.
Norwood, MA: Artec House, 1999.

[4] Y. Boers and J. N. Driessen
Particle filter based detection for tracking.
In *Proceedings of the American Control Conference*, Arlington, VA, June 2001, 4393–4397.

[5] S. B. Colegrove and S. J. Davey
PDAF with multiple clutter regions and target models.
IEEE Transactions on Aerospace and Electronic Systems, **39**, 1 (Jan. 2003), 110–124.

[6] S. J. Davey
Histogram PMHT with particles.
In *Proceedings of the 14th International Conference on Information Fusion*, July 2011.

[7] S. J. Davey and D. A. Gray
Integrated track maintenance for the PMHT via the hysteresis mdoel.
IEEE Transactions on Aerospace and Electronic Systems, **43**, 1 (Jan. 2007), 93–111.

[8] S. J. Davey, M. G. Rutten, and B. Cheung
A comparison of detection performance for several track-before-detect algorithms.
EURASIP Journal on Advances in Signal Processing, 2008.

[9] R. M. Haralick and L. G. Shapiro
Computer and Robot Vision.
Addison-Wesley, 1992, vol. 1.

[10] T. E. Luginbuhl, Y. Sun, and P. Willett
A track management system for the PMHT algorithm.
In *Proceedings of the 4th International Conference on Information Fusion*, Montreal, Canada, Aug. 2001.

[11] M. McDonald and B. Balaji
Continuous-discrete filtering for dim manoeuvring maritime targets.
July 2007.

[12] M. McDonald and B. Balaji
Impact of measurement model mismatch on nonlinear track-before-detect performance.
In *Radar Conference, 2008* (RADAR '08, IEEE), May 2008, 1–6.

[13] M. McDonald and B. Balaji
Track-before-detect using swerling 0, 1, and 3 target models for small manoeuvring maritime targets.
Journal on Advances in Signal Processing (EURASIP), 2008.

[14] R. M. Burczewski and N. C. Mohanty
Detection of moving optical objects.
In *International Telemetering Conference*, Los Angeles, CA, 1978, 325–330.

[15] E. Oron, A. Kumar, and Y. Bar-Shalom
Precision tracking with segmentation for imaging sensors.
IEEE Transactions on Aerospace and Electronic Systems, **29**, 3 (July 1993), 977–987.

[16] A. G. Pakfiliz and M. Efe
Multi-target tracking in clutter with histogram probabilistic multi-hypothesis tracker.
In *IEEE Conference on Systems Engineering*, 2005, 137–142.

[17] S. C. Pohlig
An algorithm for detection of moving optical targets.
IEEE Transactions on Aerospace and Electronic Systems, **25**, 1 (Jan. 1989), 56–63.

[18] B. Ristic, S. Arulampalam, and N. J. Gordon
Beyond the Kalman Filter: Particle Filters for Tracking Applications.
Artech House, 2004.

[19] D. J. Salmond and H. Birch
A particle filter for track-before-detect.
In *Proceedings of the American Control Conference*, Arlington, VA, June 2001, 3755–3760.

[20] M. C. Smith and E. M. Winter
On the detection of target trajectories in a mutli target environment.
In *IEEE Conference on Decision and Control*, San Diego, CA, 1978, 1189–1194.

[21] L. D. Stone, C. A. Barlow, and T. L. Corwin
Bayesian Multiple Target Tracking.
Artech House, 1999.

[22] R. L. Streit
Tracking on intensity-modulated data streams.
NUWC, RI: Technical report 11221, May 2000.

[23] R. L. Streit
Tracking targets with specified spectra using the H-PMHT algorithm.
NUWC, RI: Technical report 11291, June 2001.

[24] R. L. Streit, M. L. Graham, and M. J. Walsh
Multitarget tracking of distributed targets using histogram-PMHT.
Digital Signal Processing, **12**, 2 (July 2002).

[25] B-N. Vo, B-T. Vo, N-T. Pham, and D. Suter
Joint detection and estimation of multiple objects from image observations.
IEEE Transactions on Signal Processing, **58**, 10 (Oct. 2010), 5129–5141.

[26] M. Wieneke and S. J. Davey
Histogram PMHT with target extent estimates based on random matrices.
In *Proceedings of the 14th International Conference on Information Fusion*, July 2011.

Samuel Davey received the Bachelor of Engineering, Master of Mathematical Science, and Ph.D. degrees from the University of Adelaide, Australia, in 1996, 1999, and 2003, respectively.

Since 1995 he has worked for the Defence Science and Technology Organisation, Australia, in the areas of target tracking, tracker performance assessment, and multi-sensor fusion. He is also a Visiting Research Fellow at the University of Adelaide.

Real-time Allocation of Firing Units To Hostile Targets

FREDRIK JOHANSSON
GÖRAN FALKMAN

The protection of defended assets such as military bases and population centers against hostile targets (e.g., aircrafts, missiles, and rockets) is a highly relevant problem in the military conflicts of today and tomorrow. In order to neutralize threats of this kind, they have to be detected and engaged before causing any damage to the defended assets. We review algorithms for solving the resource allocation problem in real-time, and empirically investigate their performance using the open source testbed SWARD. The results show that many of the tested algorithms produce high quality solutions for small-scale problems. A novel variant of particle swarm optimization seeded with an enhanced greedy algorithm is described and is shown to perform best for large instances of the real-time allocation problem.

Manuscript received October 14, 2010; revised February 21, July 12, and September 9, 2011; released for publication September 29, 2011.

Refereeing of this contribution was handled by Huimin Chen.

Authors' addresses: F. Johansson, Division of Information Systems, Swedish Defence Research Agency, SE-164 90, Stockholm, Sweden, E-mail: (fredrik.johansson@foi.se); G. Falkman, Informatics Research Centre, University of Skövde, PO Box 408, SE-541 28 Skövde, Sweden, E-mail: (goran.falkman@his.se).

1. INTRODUCTION

A severe threat encountered in many international peacekeeping and peace forcing operations is that of rockets, artillery, and mortars (RAM) fired by insurgents towards military bases, troops, and other assets. Attacks like these have cost many human lives in places like Iraq and Afghanistan during recent years. Similar attacks are faced by civilians in some parts of Israel on a regular basis, where so called Katyusha and Qassam rockets are fired against Israeli population centers such as Sderot and Ashkelon. Asymmetrical threats like these have caused an increased interest in systems for detecting and tracking incoming RAM before they hit their intended targets. The detection and tracking of RAM makes it possible to estimate the point of impact, so that any troops or civilians in the impact area can be alerted. However, such a warning is not always enough, due to very quick course of events, and that buildings and infrastructure will be destroyed no matter how early warnings come, given that active countermeasures are not taken. Hence, one would like to destroy incoming RAM before they hit their intended targets (and before they risk causing collateral damage upon destruction). Systems for detecting, tracking, and engaging RAM are often referred to as Counter Rocket, Artillery, and Mortar (C-RAM) systems. An example of such a system is the recently deployed Israeli Iron Dome system. Another kind of air defense situation is that in which we would like to protect defended assets against maneuvering targets such as fighter aircrafts, attack helicopters, and non-ballistic missiles. For such kind of threats, we can in general not easily predict which defended asset (if any) is the intended target of an attack, making it necessary to estimate the level of threat posed by detected targets to the defended assets in a so-called threat evaluation process.

When faced with many simultaneous threats, it is unlikely that the defenders can take action against all incoming threats, since there often are fewer firing units available than there are threats. Even when this is not the case, a problem is to know which firing unit to use against which threat in order to maximize the survivability of the defended assets or minimize the total expected target value of surviving hostile targets. This can be described as a resource allocation problem, known as the weapon allocation problem [7] within the field of operations research. Unfortunately, the allocation of defensive firing units to targets has been shown to be **NP**-complete [23].

The time available for weapon allocation depends on many factors such as the type of RAM used, the range from which it is fired, type of detection radar and type of defensive weapons (rapid-fire guns, lasers, radar-guided missiles, etc.). However, taking into account that the incoming threats often have high speed and are fired from a range of only a few kilometers, very short time is available for detection, weapon allocation, and

interception. Hence, empirical results for how weapon allocation algorithms perform on problem instances of various size are needed.

We have reviewed the available literature in order to identify suitable weapon allocation algorithms, and we have implemented and systematically evaluated the real-time performance of a selection of the identified algorithms on static asset-based weapon allocation problems. The results show that especially particle swarm optimization algorithms produce high quality solutions for small-scale problems. In this article, we describe a novel variant of particle swarm optimization seeded with an enhanced greedy algorithm and show that the seeded version performs very well relative to previously tested algorithms also for large-scale instances of the real-time allocation problem.

The rest of this article is structured as follows. In Section 2, we present the static asset-based weapon allocation problem, which is a suitable optimization model when the impact area of a threat can be assumed to be known. We also present its target-based counterpart which is more suitable for air defense situations involving maneuvering targets. In Section 3, we present a literature survey of algorithms that have been suggested for static weapon allocation (both target-based and static-based). Based on this survey, we have implemented algorithms for static asset-based weapon allocation which are presented in Section 4. Experiments in which we compare the real-time performance of the implemented algorithms are presented in Section 5, and we conclude the article in Section 6.

2. WEAPON ALLOCATION

Informally, weapon allocation (often also referred to as weapon assignment or weapon-target allocation) can be defined as the reactive assignment of defensive weapon resources (firing units) to engage or counter identified threats (e.g., aircrafts, air-to-surface missiles, and rockets) [29]. More formally, the weapon allocation problem can be stated as a non-linear optimization problem in which we aim to allocate firing units so as to minimize the expected total value of the targets, or, alternatively, to maximize the expected survivability of the defended assets. These alternative views are referred to as target-based (weighted subtractive) defense and asset-based (preferential) defense, respectively. The asset-based formulation demands knowledge of which targets that are headed for which defended assets and thereby assumes a high level of situation awareness [24]. Therefore, the static asset-based weapon allocation problem formulation is suitable for problems involving defense against ballistic weapons, while the target-based formulation is more appropriate when the intended aims of the targets are not known [28]. In Section 2.1 we describe the static asset-based weapon allocation problem, and in Section 2.2 we give a similar description of the static target-based weapon allocation problem.

2.1. The Static Asset-Based Weapon Allocation Problem

When presenting the static asset-based weapon allocation problem, the following notation will be used:

- $|\mathbf{A}| \triangleq$ number of defended assets.
- $|\mathbf{W}| \triangleq$ number of firing units.
- $|\mathbf{T}| \triangleq$ number of targets.
- $\omega_j \triangleq$ protection value of defended asset A_j.
- $P_{ik} \triangleq$ probability that firing unit W_k destroys target T_i if assigned to it.
- $\pi_i \triangleq$ probability that target T_i destroys the asset it is aimed for.
- $\mathbf{G}_j \triangleq$ the set of targets aimed for defended asset A_j.
- $x_{ik} = \begin{cases} 1 & \text{if firing unit } W_k \text{ is assigned to target } T_i, \\ 0 & \text{otherwise.} \end{cases}$

In the static asset-based weapon allocation problem, each offensive target is assumed to be aimed at a defended asset, where each defended asset is associated with a protection value ω_j. Each target has an associated lethality probability π_i, indicating the probability that T_i destroys the defended asset it is aimed for, given that it is not successfully engaged. This probability depends on the accuracy of the targets as well as the nature of the defended assets [8]. As can be seen, we are assuming that such probabilities are target dependent only, i.e., we do not take the type of the defended asset into consideration. The defenders are equipped with firing units, where each pair of firing unit and target is assigned a kill probability P_{ik}. Now, the objective of the defense is to allocate the available firing units so as to maximize the total expected protection value of surviving defended assets [7]:

$$\max J = \sum_{j=1}^{|\mathbf{A}|} \omega_j \prod_{i \in \mathbf{G}_j} \left(1 - \pi_i \prod_{k=1}^{|\mathbf{W}|} (1 - P_{ik})^{x_{ik}} \right) \tag{1}$$

subject to:

$$\begin{aligned} \sum_{i=1}^{|\mathbf{T}|} x_{ik} &= 1, \qquad \forall k \\ x_{ik} &\in \{0,1\}, \qquad \forall i \forall k. \end{aligned} \tag{2}$$

In (1), the inner product $\prod_{k=1}^{|\mathbf{W}|}(1 - P_{ik})^{x_{ik}}$ should be interpreted as the probability that target T_i survives the countermeasures taken against it. Hence, the product $\prod_{i \in \mathbf{G}_j}(1 - \pi_i \prod_{k=1}^{|\mathbf{W}|}(1 - P_{ik})^{x_{ik}})$ is the probability that the defended asset A_j survives the attack of all targets aimed for it.

A solution to a static weapon allocation problem can be represented as a matrix of decision variables

$$\mathbf{x} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1|\mathbf{W}|} \\ x_{21} & x_{22} & \cdots & x_{2|\mathbf{W}|} \\ \vdots & \vdots & x_{ik} & \vdots \\ x_{|\mathbf{T}|1} & x_{|\mathbf{T}|2} & \cdots & x_{|\mathbf{T}||\mathbf{W}|} \end{bmatrix}. \quad (3)$$

Such a solution is feasible if it fulfills the constraints given in (2), i.e., that the entries of each column in (3) sum to one. For a problem instance consisting of $|\mathbf{T}|$ targets and $|\mathbf{W}|$ firing units, there are $|\mathbf{T}|^{|\mathbf{W}|}$ feasible solutions.

From the solution of the static asset-based weapon allocation problem, we can discover which of the defended assets that should be protected, and in which way each of the defended assets should be protected (preferential defense).

2.2. The Static Target-Based Weapon Allocation Problem

Using the same notation as in Section 2.1, but with the additional definition:

- $V_i \triangleq$ target value of target T_i,

we can define the static target-based weapon allocation problem as:

$$\min F = \sum_{i=1}^{|\mathbf{T}|} V_i \prod_{k=1}^{|\mathbf{W}|} (1 - P_{ik})^{x_{ik}} \quad (4)$$

subject to the constraints given in (2). Since the product as before is the probability that target T_i survives the countermeasures taken against it, the objective function should be interpreted as the minimization of the total expected target value of surviving targets.

The estimation of target values is far from trivial, and can be seen as a very important high-level information fusion problem. A survey of how threat values V_{ij}s can be estimated (representing the threat posed by target T_i to defended asset A_j) is presented in [11, 12]. Once such threat values have been calculated, these can be aggregated into target values using weighted averages such as:

$$V_i = \frac{\sum_{j=1}^{|\mathbf{A}|} V_{ij}\omega_j}{\sum_{j=1}^{|\mathbf{A}|} \omega_j}. \quad (5)$$

Nevertheless, this is only one choice of how to aggregate threat values into target values. Furthermore, the original target values rely on coarse models of what is threatening behavior or not (typically parameters such as distance between the target and the defended asset, the speed and heading of the target, target type, etc). To complicate matters, expert air defense operators frequently disagree about the threat of individual aircraft [31]. Consequently, it should be remembered that target values will always be associated with uncertainty, and that they to a large degree are subjective.

2.3. Properties of Weapon Allocation Problems

A few assumptions are made in the static weapon allocation formulations. Firstly, all firing units have to be assigned to targets, as indicated in the constraint given in (2). Moreover, all the firing units have to be assigned simultaneously, i.e., we can not observe the outcome of some of the engagements before a remaining subset of firing units are allocated. This is what is meant by *static* weapon allocation, as opposed to *dynamic* weapon allocation. The static formulation makes sense for the problem domain studied here, since the high speed of short-range RAM does not allow for several engagement cycles. We also assume that an engagement will not affect other engagements (e.g., that a firing unit can destroy another target than it is allocated to, or that targets can destroy other assets than they are aimed for). Without the last assumption, the geometry of the problem must be taken into account, creating an extremely complex problem. We also ignore the risk of collateral damage to the protected area when intercepting the targets.

Despite the assumptions, there is a combination of factors that make the static weapon allocation problems hard to solve. Firstly, the objective functions given in (1) and (4) are *non-linear*, so that well-known linear programming techniques such as the simplex algorithm can not be used to solve the problems. Secondly, the problems are *discrete*, since they only allow for integer valued feasible solutions due to the second constraint in (2) (i.e., fractional allocations are not possible). In general, this kind of integer programming problems are hard to solve. Thirdly, the problems are *stochastic*, due to kill probabilities (and lethality probabilities) not equal to zero or one. This non-determinism further complicates the problems. Fourthly, it is not unusual with *large-scale* problem instances, i.e., problems consisting of a large number of firing units, defended assets, and/or targets. The asset-based formulation can be shown to be a generalization of the static target-based weapon allocation formulation [7] presented in Section 2.2. The **NP**-completeness of the static target-based weapon allocation problem was established in [23], and hence, we can conclude that the static asset-based weapon allocation problem is **NP**-complete as well [7]. These properties taken together show that finding good solutions in real-time to static weapon allocation problems is indeed a very hard problem, and according to [7], rule out any hope of obtaining efficient optimal algorithms.

3. A SURVEY OF ALGORITHMS FOR WEAPON ALLOCATION

Initial research on the static target-based weapon allocation problem dates back as far as the end of the 1950s (cf. [5, 25]). Much of the initial research on the

problem seems to have been motivated by the threat from intercontinental ballistic missiles during the Cold War era [24]. Despite the end of the Cold War, research on defensive weapon allocation still remains a very active area [24]. The static target-based weapon allocation problem has been quite well studied, especially within the field of operations research. Despite the extensive research, static weapon allocation is an example of a classical operations research problem that still remains unsolved [1], in the sense that effective methods for real-time allocation are lacking. Moreover, the asset-based version of the problem is much less studied than its target-based counterpart.

Much of the original work on weapon allocation focused on the allocation of missiles to defended assets, rather than the other way around. Hence, the problems were often modeled from an attacker's side, instead of from the defending side. A brief summary and review of unclassified literature from the first years of research on the problem is given in [26]. Some years later, a monograph describing many of the developed mathematical models for weapon allocation problems was published in [6]. Unlike Matlin's review, the monograph by Eckler and Burr takes on the weapon allocation problem from a defender's view. The authors present a number of useful techniques for weapon allocation, such as relaxing the integer constraint and then make use of linear programming to solve the resulting continuous problem. This is a technique that is still in use (cf. [17]). It should be noted however, that fractional assignments of firing units to targets does not make sense, and rounding off the optimal solution to the relaxation of an nonlinear integer programming problem can yield solutions that are infeasible or far from the optimal solution to the original nonlinear problem [36]. Other kinds of tools such as the use of Lagrange multipliers and dynamic programming are also described in [6]. As the authors make clear, their focus is on analytical approaches, since it is argued that what they refer to as computer-oriented solutions give less insight into the weapon allocation problem than analytical approaches. A somewhat more recent survey of work within weapon allocation is presented in [3]. As in the earlier mentioned surveys, its focus is on analytical approaches to weapon allocation. However, it is mentioned that a shift towards various techniques such as implicit enumeration algorithms and nonlinear programming algorithms had been started at that time, since mathematical formulations of the weapon allocation problem are not generally amenable to solution in closed form [3, p. 66]. In later years, advanced computer-based techniques have been developed which are better suited for real-time weapon allocation [9]. In the following, we will focus on modern heuristic/approximate approaches, but will first present enumerative techniques, since such approaches can be very useful for special cases of static weapon allocation problems.

3.1. Exact Approaches

For small values of $|\mathbf{T}|$ and $|\mathbf{W}|$, the optimal solution to a static weapon allocation problem can easily be found by exhaustive search (also referred to as explicit enumeration), i.e., a brute-force enumeration where all feasible solutions are tested one after the other. However, as a static weapon allocation problem consists of $|\mathbf{T}|^{|\mathbf{W}|}$ feasible solutions, this is not a viable approach for air defense scenarios involving a large number of targets and firing units.

Exact polynomial time algorithms have been identified for the special case of the static target-based weapon allocation problem in which the kill probabilities of all firing units are assumed to be identical, i.e., $P_{ik} = P_i$. For this special case, the well known maximum marginal return (MMR) algorithm suggested in [5], and the local search algorithm suggested in [7] can be proven to be optimal. Some other special cases of the static target-based weapon allocation problem can be formulated as network flow optimization problems. If we assume the constraint that all firing units have kill probabilities $P_{ik} \in \{0, P_i\}$, i.e., that firing units either can or cannot reach a target, and in the former case, the kill probability only depend upon the target, the problem can be transformed into a minimum cost network flow problem with linear arc costs, for which several efficient algorithms exist [7]. A similar transformation can be done for the special case of the static target-based weapon allocation problem where we assume that $|\mathbf{T}| \leq |\mathbf{W}|$, and that at most one firing unit is to be allocated to each target. In this case, we can convert the problem into a so called transportation problem, for which efficient algorithms exist [7]. However, the general static target-based weapon allocation problem has been proved to be **NP**-complete [23], as have been discussed earlier. This also holds true for the asset-based version of the static weapon allocation problem, since this can be seen as a generalization of the static target-based version.

Another exact approach is to use branch-and-bound algorithms for finding the optimal solution. Branch-and-bound algorithms use tree representations of the solution space and are often able to prune away large subsets of feasible solutions through calculation of lower and upper bounds on different branches of the tree. In a recent article by [1], three branch-and-bound algorithms (using different lower-bound schemes) are investigated and are shown to give short computation times on average. The results are impressive, however, in theory the risk exists that the algorithm will require branching the full tree for some problem instances. This means that in worst-case, the performance of the branch-and-bound algorithm can be at least as bad as the performance of more naïve exhaustive search algorithms. Although it in practice is unlikely that this worst-case scenario will appear, it is unfortunately not possible to in advance compute an upper bound on the computational time it will

take to find the optimal solution to a problem instance when using a branch-and-bound algorithm. Hence, as can be seen in the results reported in [1], some problem instances of large size can be solved very quickly, while considerably smaller problem sizes can demand considerably more time for the optimal solution to be found. In other words, we have to rely on heuristic algorithms for large-scale problems when real-time guarantees are needed [1, 7].

3.2. Heuristic Approaches

A well-known heuristic approach for static target-based weapon allocation is the greedy maximum marginal return algorithm, originally suggested in [5]. A similar greedy algorithm is presented in [18]. Basically, the maximum marginal return algorithm works sequentially by greedily allocating firing units to the target maximizing the reduction of the expected value. It starts with allocating the first firing unit to the target for which the reduction in value is maximal, whereupon the value of the target is reduced to the new expected value. Once the first firing unit is allocated, the same procedure is repeated for the second firing unit, and so on, until all firing units have been allocated to targets. Pseudo code for the maximum marginal return algorithm is shown in Section 4.1. Obviously, the maximum marginal return algorithm is very simple and fast. This is a general advantage of greedy algorithms, but due to their greedy nature they are also very likely to end up with suboptimal solutions. Since the algorithm uses target values for choosing which target to be allocated next, it cannot be used as is for static asset-based weapon allocation. However, in [27] a number of greedy algorithms for asset-based weapon allocation are described. These algorithms basically work by approximating the asset-based problem with its target-based counterpart, by using the protection value of the defended asset to which the target is aimed for as the target value. When the problem has been approximated by a target-based problem, it is suggested that the maximum marginal return algorithm returns a solution that can be used as an approximative solution to the asset-based problem. Another suggested approach in [27] is to use the solution returned from the maximum marginal return algorithm and to apply local search on the solution so that the target allocated by one weapon can be swapped to the target allocated by another weapon, and vice versa.

Another kind of heuristic approach to a constrained version of the target-based weapon allocation problem has been suggested in [35], in which artificial neural networks are used. It is stated that solutions close to global optima are found by the algorithm, but results are only presented for a few small-scale problem instances, from which it in the authors' view is not possible to generalize. It is in [9] also argued that artificial neural network algorithms for weapon allocation sometimes are unsteady and non-convergent, leading to that obtained solutions may be both suboptimal and infeasible.

As an alternative, the use of genetic algorithms seems to be popular. Such an algorithm for static target-based weapon allocation is described in [16], while a genetic algorithm combined with local search is presented in [22] and [21]. The quality of the solutions returned by the greedy maximum marginal return algorithm presented in [18] is in [16] compared to the solutions returned by genetic algorithms. However, the algorithms are only evaluated on target-based weapon allocation problems. The standard genetic algorithm is outperformed on large-scale problem sizes, but only one problem instance is tested for each problem size, so the possibility to generalize the results can be questioned. Even though, the results seem to indicate that greedy search works better than standard genetic algorithms on large target-based problem instances. It is in [16] suggested that genetic algorithms can be seeded with the solution returned from a greedy algorithm, which seems to be a suitable approach to improve the quality of genetic algorithms on large problem sizes. In [2], a genetic algorithm combined with local search is suggested for a dynamic version of the asset-based weapon allocation problem. It is shown that local search improves the results, but that the computational time needed is increased. The effects of real-time requirements on the algorithms are not tested.

The use of ant colony optimization for target-based weapon allocation is suggested in [19, 20]. Reported results in [19] and [20] indicate that ant colony optimization algorithms perform better than standard genetic algorithms on large-scale problems, and that the algorithms can be improved upon by using local search. However, the algorithms were allowed to run for two hours, so it is unclear how this generalizes to settings with real-time requirements.

A simulated annealing algorithm for static asset-based weapon allocation is presented in [4]. Basically, simulated annealing is based on an analogy of thermodynamics with the way metals cool and anneal, in which a liquid that is cooled slowly is likely to form a pure crystal corresponding to a state of minimum energy for the metal, while a quick cooling phase is likely to result in states of higher energy levels [32]. By controlling an artificial "temperature" when making the optimization (corresponding to the minimization of energy levels), it becomes possible to escape from local minima in the hunt for the optimal solution (the purest crystal in the thermodynamics analogy). However, no evaluation of the quality of the solutions obtained by the algorithm is presented in [4], so it is unknown how good their implemented algorithm performs. Another implementation of a simulated annealing algorithm provides solutions of lower quality than ant colony optimization and genetic algorithms in a static target-based weapon allocation experiment described in [19]. The algorithms were, as describe above, allowed to run for two hours,

TABLE I
Algorithmic Approaches to Weapon Allocation

Algorithmic Approach	*References*
Branch-and-bound	[1]
Genetic algorithms	[16, 21]
Ant colony optimization	[19, 20]
Greedy algorithms	[5, 18]
VLSN	[1]
Neural networks	[35]
Particle swarm optimization	[33, 37]

so it is not known how the algorithms perform under more realistic time constraints.

In [1], good performance results for an approach using a minimum cost flow formulation heuristic for generating a good starting feasible solution are presented. This feasible solution is then improved by a very large-scale neighborhood (VLSN) search algorithm that treats the problem as a partitioning problem, in which each partition contains the set of firing units assigned to target T_i. The very-large scale neighborhood search improves the original feasible solution by a sequence of cyclic multi-exchanges and multi-exchange paths among the partitions. As the name suggests, the size of the used neighborhoods are very large. To search such large neighborhoods typically takes considerably amounts of computations and demands implicit enumeration methods [10]. By using the concept of an improvement graph, it becomes possible to evaluate neighbors faster than other existing methods [1, 10].

Recently, the use of particle swarm optimization for static target-based weapon allocation has been suggested. In [33], a particle swarm optimization algorithm is implemented and compared to a genetic algorithm. The results indicate that the particle swarm optimization algorithm generates better solutions than the genetic algorithm, but the algorithms are only tested on a single problem instance consisting of five targets and ten firing units. For this reason, it is not possible to generalize the obtained results. Experiments presented in [37] also indicate that particle swarm optimization algorithms create better solutions than genetic algorithms for static target-based weapon allocation.

As evident from the literature survey presented above, a lot of different algorithmic approaches have been suggested for the static weapon allocation problem. A summary of some of the approaches presented above is presented in Table I.

4. THE IMPLEMENTED ALGORITHMS

Based on the results from the literature survey presented in Section 3, a number of heuristic algorithms have been implemented. Since the target-based weapon allocation seems more well-researched than the asset-based problem, the focus of the rest of this article will be on the latter.

The algorithms for static asset-based weapon allocation evaluated in this article share the same kind of representation, in which a solution is represented as a vector of length $|\mathbf{W}|$. Each element k in the vector points out the target T_i to which the weapon is allocated. As an example of this, the vector $[2,3,2,1]$ represents a solution in which W_1 and W_3 are allocated to T_2, W_2 is allocated to T_3, and W_4 is allocated to T_1.

4.1. A Maximum Marginal Return Algorithm for Static Weapon Allocation

A greedy algorithm for static target-based weapon allocation, known as the maximum marginal return (MMR) algorithm, was initially suggested in [5]. This algorithm (described with pseudo code in Algorithm 1) is very simple since it as already explained works greedily by assigning weapons sequentially to the target that maximizes the reduction of the expected target value. When the first weapon has been allocated to the target for which the reduction in value is maximal, the target value is reduced to the new expected value. After that, the same procedure is repeated for the second weapon, and so on, until all weapons have been allocated to targets, yielding a computational complexity of $O(|\mathbf{W}| \times |\mathbf{T}|)$.

ALGORITHM 1 *Maximum marginal return algorithm*
for all k *such that* $1 \leq k \leq |\mathbf{W}|$ **do**
  $highestValue \leftarrow -\infty$
  $allocatedTarget \leftarrow 0$
  for all i *such that* $1 \leq i \leq |\mathbf{T}|$ **do**
    $value \leftarrow V_i \times P_{ik}$
    if $value > highestValue$ **then**
      $highestValue \leftarrow value$
      $allocatedTarget \leftarrow i$
  assign W_k *to target* $T_{allocatedTarget}$
  $V_{allocatedTarget} \leftarrow V_{allocatedTarget} - highestValue$
return allocation

It is not obvious how to use the MMR algorithm for the static asset-based weapon allocation problem, since it in this version of the problem does not exist any target values. Instead, there are protection values associated with the defended assets, and lethality values associated with the targets. In [27], it is suggested that a defended asset's weight (protection value) is equally distributed over the targets aimed for it, so that a target's value is computed as $V_i = \omega_j / |\mathbf{G}_j|$ (where j is the index for the set $\mathbf{G}_j$ of which target T_i is a member), and that the asset-based problem is approximated with its target-based counterpart. Similar reasoning is presented in [7] where it is suggested that the value of a target is set to the expected destroyed protection value of the defended asset to which it is aimed, given that the target is not engaged and that all other targets aimed for the defended asset are destroyed.

We have here chosen to calculate the target value V_i for a target T_i as:

$$V_i = \omega_j \times \pi_i \tag{6}$$

where j is the index of the defended asset to which target T_i is aimed. Hence, the target value has been calculated as the product of the lethality probability π_i of the target and the protection value ω_j of the defended asset it is aimed at. In this way, we follow the approach suggested in [7] to use the protection value of the defended asset to impact on the target value, but we complement this with taking the lethality of the target into account, since this extra information otherwise is lost.

We have also included a variant of greedy search where we have taken the solution generated by the MMR algorithm and improved it with a simple local search (LS) that creates neighbor solutions by swapping two positions selected at random in the solution vector (this variant of the MMR algorithm, described with pseudo code in Algorithm 2, will in the following be referred to as MMR-LS). This algorithm is an implementation of the idea briefly discussed in [27].

ALGORITHM 2 *The MMR-LS algorithm*

```
bestSolution ← MMR()
J_best ← CalculateFitness(bestSolution)
while termination criteria not met do
  neighborSolution ← neighbor(bestSolution)
  J_new ← CalculateFitness(neighborSolution)
  if J_new > J_best then
    bestSolution ← neighborSolution
    J_best ← J_new
return bestSolution
```

Obviously, the quality of the solutions generated by the MMR-LS algorithm will always be at least as good as the quality of the solutions returned by the MMR algorithm.

4.2. An Enhanced Maximum Marginal Return Algorithm for Static Weapon Allocation

What here will be referred to as the enhanced maximum marginal return algorithm (the authors' terminology) is quite similar to the standard maximum marginal return algorithm. The difference is that in the enhanced maximum marginal return (EMMR) algorithm it is not predetermined which firing unit to allocate next. Instead, the choice of which firing unit to allocate next is based on which weapon-target pair that maximizes the marginal return. We have implemented this algorithm based on the description in [16], and the pseudo code for the algorithm is given in Algorithm 3. In the first iteration $it = 1$, $|\mathbf{W}| \times |\mathbf{T}|$ combinations are tested. The weapon-target pair with highest marginal return is selected, so that the firing unit is selected to the target, and the target value of the corresponding target is updated accordingly. After this, $|\mathbf{W}| - 1$ firing units are unallocated. In next iteration, the remaining $(|\mathbf{W}| - 1) \times |\mathbf{T}|$ weapon-target pairs are tested, and so on, until there does not remain any unallocated firing units. Hence, the time complexity of EMMR becomes $O(|\mathbf{W}|^2|\mathbf{T}|)$.

ALGORITHM 3 *Enhanced maximum marginal return algorithm (adapted from [16])*

```
for all it such that 1 ≤ it ≤ |W| do
  highestValue ← −∞
  allocatedTarget ← 0
  allocatedWeapon ← 0
  for all k such that 1 ≤ k ≤ |W| do
    for all i such that 1 ≤ i ≤ |T| do
      value ← V_i × P_ik
      if value > highestValue then
        highestValue ← value
        allocatedWeapon ← k
        allocatedTarget ← i
  assign W_allocatedWeapon to T_allocatedTarget
  V_allocatedTarget ← V_allocatedTarget − highestValue
return allocation
```

As the standard MMR algorithm, EMMR is relying on target values. Hence, we calculate target values according to (6), solve the approximated target-based problem using EMMR, and return the solution as the solution to the asset-based problem.

4.3. A Genetic Algorithm for Static Weapon Allocation

In [13], we presented a genetic algorithm (GA) designed for real-time allocation of defensive weapon resources to targets. The original version of the algorithm was intended for the static target-based problem, but we have now with some modifications adapted it to also suit the static asset-based formulation of the problem.

The algorithm is described in pseudo code in Algorithm 4. First, an initial population consisting of *nrOfIndividuals* is created, through generation of a vector of length $|\mathbf{W}|$. In this vector each element W_k is assigned a random integer value in the interval $\{1,\ldots,|\mathbf{T}|\}$. In each generation we evaluate all individuals in the population and determine their objective function values in accordance with (1). Hence, each individual is assigned a fitness value that is used in the following phases of selection and recombination. After the evaluation phase, deterministic tournament selection is used as selection mechanism to determine which individuals in population *Pop* that should be used as parents for *Pop*′, i.e., we pick two individuals at random from *Pop* and select the one with best fitness value. When two parents have been selected from *Pop*, we apply one-point crossover at a randomly selected position $k \in \{1,\ldots,|\mathbf{W}|\}$, generating two individuals that become members of *Pop*′. This is repeated until there are *nrOfIndividuals* in *Pop*′. Thereafter, we apply mutation on a randomly selected position $k \in \{1,\ldots,|\mathbf{W}|\}$ in the first individual of *Pop*′, where the old value is changed into $i \in \{1,\ldots,|\mathbf{T}|\}$. Hence, there is a probability of $1/|\mathbf{T}|$ that the individual is unaffected of the mutation. The mutation operator is repeated on all individuals in *Pop*′ and the resulting individuals become members of the new population *Pop*. This loop is repeated until the termination criterion is fulfilled (the upper limit on

the computational time bound is reached). At this point, the individual with the best fitness found during all generations is returned as the allocation recommended by the algorithm.

ALGORITHM 4 *Pseudo code for our genetic algorithm*

```
fitness_best ← −∞
Pop ← GenerateInitialPopulation()
while termination criteria not met do
  for l ← 1 to nrOfIndividuals do
    J_l ← CalculateFitness(Pop(l))
    if J_l > fitness_best then
      g⃗ ← Pop(l)
      fitness_best ← J_l
  Pop′ ← Crossover(Pop)
  Pop ← Mutate(Pop′)
return g⃗
```

Furthermore, we have implemented a variant of the genetic algorithm that is seeded with well-performing individuals. Instead of creating all individuals in the initial population at random, κ individuals are created based on the solution returned by the EMMR algorithm (a random swap between the targets of two of the firing units is first made for each of the seeded individuals in order to create some diversity among them). The remaining individuals are created randomly just as before. This seeded version of the genetic algorithm will in the following be referred to as GA-S.

4.4. A Particle Swarm Optimization Algorithm for Static Weapon Allocation

In [14], we developed a particle swarm optimization (PSO) algorithm for the static target-based weapon allocation problem. We have modified this algorithm to also suit the static asset-based weapon allocation problem.

A particle swarm consists of *nrOfParticles* particles, in which each particle is associated with a position $\vec{x}_l^t$, a velocity $\vec{v}_l^t$, and a memory $\vec{b}_l^t$ storing the particle's personal best position. Moreover, we also store the swarm's global best position in a vector $\vec{g}^{\,t}$. Each particle corresponds to a solution, given by the particle's position.

ALGORITHM 5 *Pseudo code for our particle swarm optimization algorithm*

```
Initialization()
while termination criteria not met do
  for l ← 1 to nrOfParticles do
    J_l ← CalculateFitness(x⃗_l)
    if J_l = CalculateFitness(g⃗) then
      p_l ← Reinitialize()
    else
      if J_l > CalculateFitness(b⃗_l) then
        b⃗_l ← x⃗_l
        if J_l > CalculateFitness(g⃗) then
          g⃗ ← x⃗_l
  for l ← 1 to nrOfParticles do
    v⃗_l ← UpdateVelocity(p⃗_l)
    x⃗_l ← UpdatePosition(p⃗_l)
return g⃗
```

The algorithm is described in pseudo code in Algorithm 5. In an initialization phase, each particle is assigned an initial position $\vec{x}_l^0$ (where the elements in the initial position vectors are integers randomly distributed between 1 and $|\mathbf{T}|$), and an initial velocity $\vec{v}_l^0$ (a vector of real numbers randomly distributed from the uniform distribution $U[-0.5|\mathbf{T}|, 0.5|\mathbf{T}|]$). A fitness value is calculated for each particle, given by the objective function value J (see (1)) that is obtained for the solution corresponding to the particle's position. The new fitness is compared to the personal best and the global best to see whether these should be updated accordingly. After this, the velocity and position is updated for each particle, according to (7) and (8).

$$\vec{v}_l^{\,t+1} = \omega\vec{v}_l^{\,t} + c_1\vec{r}_1^{\,t} \circ (\vec{b}_l^{\,t} - \vec{x}_l^{\,t}) + c_2\vec{r}_2^{\,t} \circ (\vec{g}^{\,t} - \vec{x}_l^{\,t}) \quad (7)$$

$$\vec{x}_l^{\,t+1} = \vec{x}_l^{\,t} + \vec{v}_l^{\,t+1}. \quad (8)$$

In (7), ω is a parameter referred to as inertia or momentum weight, specifying the importance of the previous velocity vector, while c_1 and c_2 are positive constants specifying how much a particle should be affected by the personal best and global best positions (referred to as the cognitive and social components, respectively). $\vec{r}_1^{\,t}$ and $\vec{r}_2^{\,t}$ are vectors with random numbers drawn uniformly from the interval $[0, 1]$. Moreover, the $\circ$-operator denotes the Hadamard product, i.e., element-by-element multiplication of the vectors. In order to avoid that particles gain too much momentum, a $V_{\max}$ parameter that constrains the velocities to stay in the interval $[-V_{\max}, V_{\max}]$ has been introduced.

After the position update specified in (8), we round off the particles' positions to their closest integer counterpart. In next iteration we calculate the particles' new fitness values, whereupon the velocities and positions are updated, and so on. This is repeated until a termination criterion is met, i.e., that no more time remains. When this happens, the best solution obtained so far is returned as output from the algorithm.

A problem that must be handled is particles moving outside the bounds of the search space. When this happens, we reinitialize the position and velocity values of the coordinate for which the problem occurred. Moreover, in order to avoid premature convergence to local optima (stagnation), we reinitialize the velocity vector for particles rediscovering the current best solution. For a more thorough explanation of the problem of stagnation in particle swarm optimization, see [34].

In addition to the described particle swarm optimization algorithm, we have also included a variant in which we seed the starting position for κ particles in the initial population in the same way as with the GA-S algorithm (while their initial velocities are randomized in the same manner as for the remaining particles). This seeded particle swarm optimization algorithm will in the following be referred to as PSO-S. To the best of our knowledge, the use of seeded particles is novel for the weapon allocation problem.

5. EXPERIMENTS

In the experiments reported here, we have used the open source testbed SWARD[1] (System for Weapon Allocation Research and Development) which we have developed in order to allow for systematic comparison of various weapon allocation algorithms [11, 15]. The testbed is implemented in Java, and we have been running the experiments on a computer with a 2.67 GHz Intel Core i7 CPU and 8 GB RAM. By using SWARD, we make sure that the experiments presented here are easily reproducible, so that researchers can test other algorithms on the same problem instances.

In order to recreate the problem instances used in the experiment presented in Section 5.1.1, the following settings should be used in SWARD:

- $T_{start} = 5$, $W_{start} = 5$,
- $T_{end} = 9$, $W_{end} = 9$,
- $T_{step} = 1$, $W_{step} = 1$,
- *iterations* = 10, *DAs* = 5,
- *seed* = 0, *timeLimit* = 1000 ms.

Similarly, for recreating the problem instances used in the experiment presented in Section 5.1.2, the following settings should be used:

- $T_{start} = 10$, $W_{start} = 10$,
- $T_{end} = 30$, $W_{end} = 30$,
- $T_{step} = 10$, $W_{step} = 10$,
- *iterations* = 100, *DAs* = 5,
- *seed* = 0, *timeLimit* = 1000 ms.

The time limits make sure that no algorithms are allowed to run for more than a total time of one second (including seeding). For the genetic algorithms we have used the parameter setting: $nrOfIndividuals = \max(|\mathbf{T}|, |\mathbf{W}|)$. Additionally, we have for the seeded version used $\kappa = 0.5 \times nrOfIndividuals$. For the particle swarm optimization algorithm we have used $nrOfParticles = 50$, $c_1 = 2.0$, $c_2 = 2.0$, $\omega = 0.8$, and $V_{\max} = 0.5 \times |\mathbf{T}|$. The same settings have been used for the seeded version, with the additional parameter setting $\kappa = 25$.

5.1. Heuristic Algorithm Performance

For scenarios that demand solving the static asset-based weapon allocation problem faster than is possible with optimal algorithms, we have to rely on heuristic algorithms. In Section 5.1.1, we present experimental results obtained with the suggested heuristic algorithms on small-scale problem instances, while we in Section 5.1.2 present results on large-scale problem instances.

5.1.1. A comparison against the optimal solution for small-scale problems

We have in order to investigate the quality of the solutions generated by the suggested algorithms compared their obtained objective function values to the optimal objective function values obtained by exhaustive search for relatively small-scale scenarios between ($|\mathbf{T}| = 5, |\mathbf{W}| = 5$) and ($|\mathbf{T}| = 9, |\mathbf{W}| = 9$).

TABLE II
Deviation from Optimal Solution (in %)
Averaged Over Ten Problem Instances

	5×5	6×6	7×7	8×8	9×9
GA	**0**	**0**	**0**	0.1	0.7
GA-S	**0**	**0**	**0**	0.2	0.5
PSO	**0**	**0**	**0**	**0**	**0.2**
PSO-S	**0**	0.1	0.1	0.2	**0.2**
MMR	2.9	3.7	4.8	6.4	6.6
EMMR	0.3	0.8	0.8	0.8	0.9
MMR-LS	0.6	1.1	0.8	1.3	1.7

The average percentage deviation from the optimal solution is a common metric to use for evaluating heuristic algorithms on small-scale optimization problems where the optimal solution can be calculated, and therefore it also has been used here. The percentage deviation Δ_{alg} for a specific algorithm on a specific problem instance has been calculated as:

$$\Delta_{alg} = \frac{|J_{alg} - J_{opt}|}{J_{opt}} \times 100 \tag{9}$$

where J_{alg} is the objective function value for the tested algorithm and J_{opt} is the optimal objective function value. In the tables, we use **bold** to show which obtained objective function value that is the best for each tested problem size.

Looking at Table II, the algorithms' percentage deviations from the optimal solution show that most of the algorithms are able to find optimal or very near-optimal solutions for the smallest tested problem sizes. The MMR algorithm is by far the worst of the algorithms on the tested small-scale scenarios, but when allowed to improve its initial solution by local search (i.e., the MMR-LS algorithm), the quality is improved. The EMMR algorithm produces solutions that are better than both the MMR and MMR-LS algorithms. However, as can be seen, all these greedy heuristics are outperformed by the nature-inspired metaheuristics. Of the nature-inspired metaheuristics, the PSO algorithm performs somewhat better than the others. In fact, it produces optimal solutions to all problem instances of size ($|\mathbf{T}| = 5, |\mathbf{W}| = 5$)–($|\mathbf{T}| = 8, |\mathbf{W}| = 8$), and for ($|\mathbf{T}| = 9, |\mathbf{W}| = 9$) it is in one second able to generate almost optimal solutions to problems consisting of $9^9 = 387{,}420{,}489$ feasible solutions.

It should be noted that the results obtained on small-scale problems do not necessarily extends to large-scale problems. For small instances of any combinatorial problem, it is likely that algorithms such as PSO algorithms and GAs are able to search a large fraction of the solution space in a short period of time, making it more probable to find a high quality solution, while

[1]The open source testbed SWARD can be downloaded from http://sourceforge.net/projects/sward/.

TABLE III
Average Objective Function Value for $|\mathbf{T}| = 10$
Averaged Over 100 Static Asset-Based Problem Instances
(higher objective function values are better)

	10×10	10×20	10×30
GA	266.7 (42.0)	301.9 (49.1)	302.4 (46.5)
GA-S	268.7 (42.3)	304.1 (49.5)	303.1 (46.6)
PSO	269.2 (42.1)	303.2 (49.4)	302.3 (46.5)
PSO-S	**270.0** (42.2)	**304.3** (49.5)	**303.2** (46.6)
MMR	251.5 (40.3)	296.8 (48.6)	301.2 (46.3)
EMMR	268.1 (42.2)	304.1 (49.5)	303.1 (46.6)
MMR-LS	267.5 (42.7)	303.6 (49.4)	302.9 (46.6)

TABLE IV
Average Objective Function Value for $|\mathbf{T}| = 20$
Averaged Over 100 Static Asset-Based Problem Instances
(higher objective function values are better)

	20×10	20×20	20×30
GA	153.7 (30.6)	219.3 (36.4)	263.5 (34.7)
GA-S	154.7 (30.9)	237.3 (37.1)	279.8 (36.0)
PSO	158.1 (30.6)	212.6 (37.2)	245.1 (34.1)
PSO-S	**160.0** (31.1)	**238.4** (37.5)	**280.0** (36.0)
MMR	117.6 (28.6)	210.2 (36.4)	261.2 (34.0)
EMMR	127.6 (29.8)	237.0 (37.0)	279.8 (36.0)
MMR-LS	128.9 (30.2)	234.9 (37.6)	277.2 (35.4)

one wrong decision by a constructive, one-pass heuristic may result in a solution differing dramatically from the optimum of a small case [30]. Therefore, the results should not without further tests be generalized to larger problem sizes. With this said, it is still very relevant to test the performance on small-scale problem instances, not at least since it in many real-world air defense scenarios is likely that the number of targets and available firing units will be close to the settings tested here.

5.1.2. A comparison between algorithms on larger-scale problems

In a second experiment with the heuristic algorithms, we have tested them on larger-scale problems ranging in between ($|\mathbf{T}| = 10, |\mathbf{W}| = 10$) and ($|\mathbf{T}| = 30, |\mathbf{W}| = 30$). The algorithms have also in this experiment been allowed to run for one second on each problem instance. The optimal solutions are hard to obtain for large-scale problem instances, so instead of calculating the deviation from the optimal solution, we have here simply plotted the objective function values obtained (averaged over 100 problem instances) in Tables III–V. We also show the associated standard deviations within parentheses. As before, **bold** is used to indicate the best obtained objective function value on each problem size.

A note to make is that the standard deviations shown in many cases are larger than the differences in mean values among the algorithms. However, this should not be interpreted as that there are no significant differences among the algorithms. Rather, the largest part of these standard deviations are due to the differences between various problem instances. In some problem instances

TABLE V
Average Objective Function Value for $|\mathbf{T}| = 30$
Averaged Over 100 Static Asset-Based Problem Instances
(higher objective function values are better)

	30×10	30×20	30×30
GA	96.8 (20.4)	147.9 (30.4)	186.2 (33.7)
GA-S	97.2 (20.3)	151.8 (29.1)	**208.7** (36.9)
PSO	99.8 (21.4)	128.6 (29.2)	150.7 (29.2)
PSO-S	**103.0** (19.9)	**155.5** (31.3)	208.6 (36.8)
MMR	53.6 (20.4)	117.0 (27.0)	174.8 (32.2)
EMMR	59.7 (24.2)	135.7 (30.6)	208.1 (36.5)
MMR-LS	59.8 (23.9)	137.6 (31.1)	205.7 (37.0)

the optimal objective function values are lower, while they in others are higher (as a natural result of the random fashion in which the problem instances are generated). As a consequence of this, also optimal algorithms would obtain large standard deviations.

When analyzing the obtained results, it can be seen that the use of local search significantly improves the quality of the solutions found using MMR also on large problem sizes. A comparison of the solutions generated by MMR-LS with the ones returned by the EMMR algorithm shows that the performance of these are approximately equally good (although EMMR is significantly faster than MMR-LS). This indicates that it in the future may be worth studying if it would be beneficial to apply simple local search also to EMMR.

It can be seen that the seeded particle swarm optimization algorithm (i.e., PSO-S) is performing best relative to the other algorithms on all tested problem sizes except the largest, on which the seeded genetic algorithm (GA-S) performs slightly better. We have in earlier work [14] shown that PSO runs into some trouble when applied to large target-based problem instances under tight real-time constraints, and this trend can be seen also for the large asset-based problem instances tested here. However, when combined with the seeding mechanism, particle swarm optimization seems to work very well. It can be seen that the obtained objective function values for the greedy algorithms MMR-LS and EMMR are reasonably close to the best algorithms' objective function values for many of the tested problem sizes, while they for problem instances where $|\mathbf{T}| > |\mathbf{W}|$ are much worse. These results are in line with the analytical arguments in [7], predicting that it will work well to approximate the static asset-based weapon allocation problem with its target-based counterpart on problem instances involving a strong defense (a large number of firing units compared to the number of targets), while the approximation will work bad in cases of a weak defense (i.e., problem instances where there are more targets than firing units). Although the differences between e.g., EMMR and PSO-S or GA-S and PSO-S are not very large for problem instances involving a strong defense, the differences should not be ignored, since such small but significant differences can have severe

impact on the end result if such algorithms are applied in a real-world C-RAM system.

6. CONCLUSIONS AND FUTURE WORK

We have in this article presented the static asset-based weapon allocation problem, which is an optimization problem that needs to be solved in a short amount of time in air defense situations involving RAM threats such as rockets and mortars. We have also presented the static target-based weapon allocation problem, but the focus has been on the asset-based case. We have implemented two versions of a genetic algorithm, two versions of a particle swarm optimization algorithm, and various versions of the greedy maximum marginal return algorithm. Such algorithms have earlier been used for the static target-based weapon allocation problem, but as far as we know, it is previously unknown how they perform on the asset-based version of the problem. Our experiments have shown that optimal or very near-optimal solutions are obtained in real-time by the genetic algorithms and the particle swarm optimization algorithms on small-scale problems. The standard maximum marginal return algorithm yields worse solutions, but these can easily be improved upon by local search, or by using an enhanced version of the algorithm. However, the quality does not become as good as that of the genetic algorithms or the particle swarm optimization algorithms.

For larger problem instances the optimal solutions are not known, and can therefore not be used for comparison. Instead, the objective function values produced by the algorithms have been compared to each other. It has been shown that the greedy algorithms create solutions of good quality (compared to the other algorithms) for scenarios with a strong defense, but that they perform bad on scenarios involving a weak defense, i.e., where there is a larger number of targets than there are firing units.

The algorithm that has been performing the best on large-scale problem instances is a novel improvement on the particle swarm optimization algorithm where the initial population is seeded with individuals based on small variations of the solution returned by the enhanced maximum marginal return algorithm. For the problem instances where there is a strong defense, the algorithm is not able to improve very much on the solution returned by the enhanced maximum marginal return algorithm, but for the problem instances involving a weak defense, the difference is dramatic. For problems of quite small scale, the difference in solution quality is very small between the particle swarm optimization algorithm and its seeded version. However, as the problem size increases, the difference in solution quality becomes very evident.

6.1. Future Work

The obtained results can be used as benchmarks for other heuristic algorithms. Hence, it is our hope that the used data sets (problem instances) will be used by other researchers as well, so that a better understanding of which algorithms that work well for static asset-based weapon allocation is obtained. Moreover, in the current research on static asset-based weapon allocation, it is assumed that kill probabilities, lethality probabilities, and target aims are known with certainty. Obviously, these estimates will in real-world systems be associated with uncertainty, and it would therefore be interesting and useful to know how sensitive the solutions produced by the algorithms are to such uncertainties.

In the experiments presented in this article, we have been generating problem instances in which there are no dependences among the values of the parameters. As an example, there is no correlation between any of the kill probabilities involving a specific target (or rather, there might be such correlation, but if so, this is by pure chance). This is consistent with how weapon allocation algorithms have been evaluated earlier in reported literature, but it can be discussed whether this lack of structure really would be seen in estimated kill probabilities from real-world air defense scenarios. Thinking of such a scenario, two targets, T_1 and T_2, of the same type, approaching a firing unit W_1 from the same direction and on the same altitude, would most likely result in kill probabilities P_{11} and P_{12} being quite similar. Likewise, two firing units W_2 and W_3 would obtain kill probabilities of approximately same magnitude, given that the firing units were positioned close together and being of the same type. Hence, the random fashion in which problem instances have been generated here (and in previous reported experiments with weapon allocation algorithms) may not necessarily create the same kinds of search spaces that would be experienced in real-world air defense situations. An idea that could be of interest for the future is therefore to create problem instances with an inbound structure that better reflect reality.

ACKNOWLEDGMENT

We would like to express gratitude to the reviewers for their constructive comments and suggestions that have helped to improve the article. This work was supported by the Information Fusion Research Program (University of Skövde, Sweden) in partnership with Saab AB and the Swedish Knowledge Foundation under grant 2003/0104.

REFERENCES

[1] R. Ahuja, A. Kumar, K. Jha, and J. Orlin
Exact and heuristic methods for the weapon target assignment problem.
Operations Research, **55**, 6 (2007), 1136–1146.

[2] J. Chen, B. Xin, Z. Peng, L. Dou, and J. Zhang
Evolutionary decision-makings for the dynamic weapon-target assignment problem.
Science in China Series F: Information Sciences, **52**, 11 (2009), 2006–2018.

[3] C. K. Cheong
Survey of investigations into the missile allocation problem.
Master's thesis, Naval Postgraduate School, Monterey, CA. 1985.

[4] C. Ciobanu and G. Marin
On heuristic optimization.
An. Stiint. Univ. Ovidius Constanta, **9**, 2 (2001), 17–30.

[5] G. G. den Broeder, R. E. Ellison, and L. Emerling
On optimum target assignments.
Operations Research, **7**, 3 (1959), 322–326.

[6] A. R. Eckler and S. A. Burr
Mathematical Models of Target Coverage and Missile Allocation.
Technical Report DTIC: AD-A953517, Military Operations Research Society, Alexandria, VA, 1972.

[7] P. A. Hosein
A Class of Dynamic Nonlinear Resource Allocation Problems.
Ph.D. thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 1990.

[8] P. A. Hosein and M. Athans
Preferential defense strategies: Part 1—the static case.
Technical report, Massachusetts Institute of Technology, 1990.

[9] C. Huaiping, L. Jingxu, C. Yingwu, and W. Hao
Survey of the research on dynamic weapon-target assignment problem.
Journal of Systems Engineering and Electronics, **17**, 3 (2006), 559–565.

[10] K. C. Jha
Very large-scale neighborhood search heuristics for combinatorial optimization problems.
Ph.D. thesis, University of Florida, 2004.

[11] F. Johansson
Evaluating the performance of TEWA systems.
Ph.D. thesis, Örebro University, 2010.

[12] F. Johansson and G. Falkman
A comparison between two approaches to threat evaluation in an air defense scenario.
In *Proceedings of the 5th International Conference on Modeling Decisions for Artificial Intelligence*, 2008, 110–121.

[13] F. Johansson and G. Falkman
An empirical investigation of the static weapon-target allocation problem.
In *Proceedings of the 3rd Skövde Workshop on Information Fusion Topics*, 2009.

[14] F. Johansson and G. Falkman
A suite of metaheuristic algorithms for static weapon-target allocation.
In *Proceedings of the 2010 International Conference on Genetic and Evolutionary Methods*, 2010.

[15] F. Johansson and G. Falkman
SWARD: System for weapon allocation research & development.
In *Proceedings of the 13th International Conference on Information Fusion*, 2010.

[16] B. A. Julstrom
String- and permutation-coded genetic algorithms for the static weapon-target assignment problem.
In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2009.

[17] O. Karasakal
Air defense missile-target allocation models for a naval task group.
Computers and Operations Research, **35**, 6 (2008), 1759–1770.

[18] S. E. Kolitz
Analysis of a maximum marginal return assignment algorithm.
In *Proceedings of the 27th Conference on Decision and Control*, 1988.

[19] Z-J. Lee and C-Y. Lee
A hybrid search algorithm with heuristics for resource allocation problem.
Information Sciences, **173**, 1–3 (2005), 155–167.

[20] Z-J. Lee, C-Y. Lee, and S-F. Su
Parallel ant colonies with heuristics applied to weapon-target assignment problems.
In *Proceedings of the 7th Conference on Artificial Intelligence and Applications*, 2002.

[21] Z. J. Lee and W. L. Lee
A hybrid search algorithm of ant colony optimization and genetic algorithm applied to weapon-target assignment problems.
In *Proceedings of the 4th International Conference on Intelligent Data Engineering and Automated Learning*, 2003, 278–285.

[22] Z. J. Lee, S. F. Su, and C. Y. Lee
A genetic algorithm with domain knowledge for weapon-target assignment problems.
Journal of the Chinese Institute of Engineers, **25**, 3 (2002), 287–295.

[23] S. P. Lloyd and H. S. Witsenhausen
Weapon allocation is NP-complete.
In *Proceedings of the 1986 Summer Conference on Simulation*, 1986.

[24] W. P. Malcolm
On the character and complexity of certain defensive resource allocation problems.
Technical Report DSTO-TR-1570, DSTO, 2004.

[25] A. S. Manne
A target-assignment problem.
Operations Research, **6**, 3 (May–June 1958), 346–351.

[26] S. Matlin
A review of the literature on the missile-allocation problem.
Operations Research, **18**, 2 (1970), 334–373.

[27] W. A. Metler and F. L. Preston
A suite of weapon assignment algorithms for a SDI midcourse battle manager.
Technical report, Naval Research Laboratory, 1990.

[28] R. A. Murphey
Target-based weapon target assigment problems.
In P. M. Pardalos and L. S. Pitsoulis (Eds.), *Nonlinear assignment problems: algorithms and applications*, 2000, 39–53.

[29] S. Paradis, A. R. Benaskeur, M. Oxenham, and P. Cutler
Threat evaluation and weapons allocation in network-centric warfare.
In *Proceedings of the 8th International Conference on Information Fusion*, 2005.

[30] R. L. Rardin and R. Uzsoy
Experimental evaluation of heuristic optimization algorithms: A tutorial.
Journal of Heuristics, 7 (2001), 261–304.

[31] M. St. John, D. I. Manes, H. S. Smallman, B. Feher, and J. G. Morrison
An intelligent threat assessment tool for decluttering naval air defense displays.
Technical report, SSC San Diego, CA, 2004.

[32] B. Suman and P. Kumar
A survey of simulated annealing as a tool for single and multiobjective optimization.
Journal of the Operational Research Society, 57 (2006), 1143–1160.

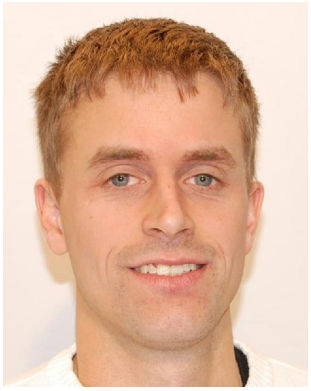
[33] P. Teng, H. Lv, J. Huang, and L. Sun
Improved particle swarm optimization algorithm and its application in coordinated air combat missile-target assignment.
In *Proceedings of the 7th World Congress on Intelligent Control and Automation*, 2008.

[34] F. van den Bergh and A. P. Engelbrecht
A new locally convergent particle swarm optimiser.
In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2002.

[35] E. Wacholder
A neural network-based optimization algorithm for the static weapon-target assignment problem.
ORSA Journal on Computing, **1**, 4 (1989), 232–246.

[36] W. L. Winston
Operations Research: Applications and Algorithms.
Wadsworth Publishing Company. 1997.

[37] X. Zeng, Y. Zhu, L. Nan, K. Hu, B. Niu, and X. He
Solving weapon-target assignment problem using discrete particle swarm optimization.
In *Proceedings of the 6th World Congress on Intelligent Control and Automation*, 2006.



Fredrik Johansson obtained his M.Sc. in computer science from University of Skövde, Sweden, in 2005 and his Ph.D. in computer science from Örebro University, Sweden, in 2010.

During his Ph.D. studies he was a member of the Skövde Artificial Intelligence Lab (SAIL) and the Information Fusion Research Program at the Informatics Research Centre (IRC) in Skövde. Currently, he works as a scientist at the Swedish Defence Research Agency (FOI) in Kista.

His research interests are applied artificial intelligence and high-level information fusion, and the application of probabilistic techniques such as Bayesian networks for decision support. He is also interested in the use of techniques such as social network analysis, natural language processing, and web harvesting for supporting the work of intelligence analysts.

Göran Falkman obtained his Ph.D. in computing science from Chalmers University of Technology, Sweden, in 2003.

He holds a position as an Associate Professor of Computer Science, with a specialty in Interactive Knowledge Systems, at University of Skövde, Sweden, where he works as a researcher and senior lecturer within the Skövde Artificial Intelligence Lab (SAIL) at the Informatics Research Centre (IRC). He has been a project leader for three applied research projects within the area of information fusion, focusing on algorithms for threat evaluation and weapon allocation, visual analytics and maritime domain awareness, and anomaly detection for surveillance applications, respectively. He has also been the leader for the Situation Awareness scenario within the Infofusion research program at University of Skövde. Currently, he is one of the principal investigators of the Uncertainty Management in High-Level Information Fusion (UMIF) research project. Since 2009, he is an elected member of the Executive Board of the Swedish Artificial Intelligence Society (SAIS).

The research interests lie in the intersection of applied artificial intelligence, knowledge systems, interaction design, and information fusion. This includes work on the design, implementation and use of formal knowledge representation and knowledge-based systems (especially, case-based reasoning, ontology engineering, and the Semantic Web), as well as the use of interactive visualization for supporting knowledge-based reasoning processes (especially, situation analysis and decision-making).

INTERNATIONAL SOCIETY OF INFORMATION FUSION

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

2011 BOARD OF DIRECTORS*

2009–2011	2010–2012	2011–2013
Éloi Bossé	Simon Maskell	Sten F. Andler
Uwe D. Hanebeck	Peter Willett	Yvo Boers
Roy Streit	Wolfgang Koch	Lyudmila Mihaylova

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

PAST PRESIDENTS

Stefano Coraluppi, 2010	Pierre Valin, 2006	Pramod Varshney, 2001
Elisa Shahbazian, 2009	W. Dale Blair, 2005	Yaakov Bar-Shalom, 2000
Darko Musicki, 2008	Chee Chong, 2004	Jim Llinas, 1999
Erik Blasch, 2007	Xiao-Rong Li, 2003	Jim Llinas, 1998
	Yaakov Bar-Shalom, 2002	

SOCIETY VISION

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

SOCIETY MISSION

Advocate

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

Serve

To serve its members and engineering, business, and scientific communities by providing high-quality information, educational products, and services.

Communicate

To create international communication forums and hold international conferences in countries that provide for interaction of members of fusion communities with each other, with those in other disciplines, and with those in industry and academia.

Educate

To promote undergraduate and graduate education related to information fusion technologies at universities around the world. Sponsor educational courses and tutorials at conferences.

Integrate

Integrate ideas from various approaches for information fusion, and look for common threads and themes–look for overall principles, rather than a multitude of point solutions. Serve as the central focus for coordinating the activities of world-wide information fusion related societies or organizations. Serve as a professional liaison to industry, academia, and government.

Disseminate

To propagate the ideas for integrated approaches to information fusion so that others can build on them in both industry and academia.

Call for Papers

The Journal of Advances in Information Fusion (JAIF) seeks original contributions in the technical areas of research related to information fusion. Authors of papers in one of the technical areas listed on the inside cover of JAIF are encouraged to submit their papers for peer review at http://jaif.msubmit.net.

Call for Reviewers

The success of JAIF and its value to the research community is strongly dependent on the quality of its peer review process. Researchers in the technical areas related to information fusion are encouraged to register as a reviewer for JAIF at http://jaif.msubmit.net. Potential reviewers should notify via email the appropriate editors of their offer to serve as a reviewer.

Journal of Advances in Information Fusion Volume 6 Number 2 December 2011 ISSN 1557-6418