December 2006 Volume 1 Number 2 ISSN 1557-6418

JAIF 06

Journal of Advances in Information Fusion

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

Regular Papers Page

From the ISIF Vice-President for Publications

The Business Model for JAIF...

ISIF

INTERNATIONAL SOCIETY OF INFORMATION FUSION

The International Society of Information Fusion (ISIF) seeks to be 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 Information 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 California Nonprofit Public Benefit Corporation at P.O. Box 4631, Mountain View, California 94040.

From the Vice President for Publications:

December 2006

The Business Model for JAIF

The *Journal for Advances in Information Fusion* (JAIF) was started following the announcement in 2002 at the International Conferences on Information Fusion (ICIF or FUSION Conference) in Annapolis, Maryland. Like every system start, it had a time delay, which in this case was more than anticipated. The first issue was published this year and a hard copy of it was distributed as part of the registration package at the FUSION Conference.

The plan was to have an on-line journal, which is what the ISIF flagship publication is. The primary credit for making this plan into reality goes to W. Dale Blair who, with his extensive experience as Editor-in-Chief of the *IEEE Transactions on Aerospace and Electronic Systems*, managed to put together the publication mechanism and orchestrate, together with Robert Lynch and Mahendra Mallick, the operation of the Editorial Board consisting of seven Area Editors and eight Associate Editors.

However, there is more to it.

The business model for this journal, as decided by the ISIF Board of Directors, is, using the terminology from the time I used to be in control, "zero-high-zero". This can be translated as follows: "no (input) fee for authors, no (output) fee for readers, only strict quality control (the transfer function—a high-pass filter)". In other words, the journal, posted on the ISIF website, is open access to anyone and, unlike other free e-journals, it will not charge the authors any publication fee. Recently, I had the experience of having a paper published in an open access e-journal, but the fee we had to pay was US$1,500.00 (this is not a typo).

The third part of the business model is to ensure the high standards of this flagship publication, as appropriate for an archival publication that is worth of this designation. While it is easier said than done, this requires a dedicated Editorial Board as well as reviewers, who can operate so that the manuscripts submitted are refereed by qualified reviewers and this should happen in a timely manner.

I want to take this opportunity to thank the entire Editorial Board (all volunteers) for their work and solicit additional volunteers for both editorial work as well as reviewers.

Yaakov Bar-Shalom
Vice President for Publications
International Society for Information Fusion

Level I and Level II Target Valuations for Sensor Management

K. C. CHANG
George Mason University

JOE P. HILL
SRS Technologies

Advanced optimization-based algorithms for sensor resource management have been the research focus area in multisensor tracking and fusion in the last decade. These algorithms for the most part offer the potential for automating the sensor control process in response to level 1 sensor data fusion (object or track-level) estimates. However, previous studies have indicated that these types of sensor resource management algorithms may have limited value in certain operational scenarios involving multi-platform surveillance and strike missions because the response is optimized for track maintenance without any assessment of overall situation context. In this paper, we will develop a framework for representing the expected information value of planned sensor measurements as it contributes to higher-level situation inferences. Specifically, a hierarchical target valuation model that estimates target value on the basis of not only a level 1 valuation function but also on the basis of a level 2 valuation function will be presented. These algorithms will provide for improved tracking and classification performance when identifying higher-level units such as convoys of vehicles. The valuation models rely on a computationally efficient implementation of Bayesian modeling and inference algorithms. Note that the main focus of the paper is on developing a hierarchical cost function that captures both level 1 and level 2 objectives and is not on developing sophisticated techniques for optimizing this objective. Simulation results which validate the approach are also presented.

Manuscript submitted on January 7, 2005; revised November 18, 2005 and September 25, 2006; released for publication February 17, 2007.

Refereeing of this contribution was handled by Dr. Chee-Yee Chong.

Authors' addresses: K. C. Chang, SEOR Department, George Mason University, 4400 University Drive, Fairfax, VA 22030, E-mail: (kchang@gmu.edu); J. P. Hill, SRS Technologies, 15000 Conference Center Drive, Suite 510, Chantilly, VA 220151, E-mail: (joe.hill@wg.srs.com).

1. INTRODUCTION

Current military Intelligence Surveillance and Reconnaissance (ISR) systems employ agile, multi-mode sensors, which are capable of producing variable scan patterns within a surveillance region in response to external tasking [18]. A number of platforms are currently available to carry out these missions. For example, long-range surveillance is accomplished with aircraft capable of high coverage rate, high signal to noise moving target indicator (MTI) and high range resolution (HRR) modes (see Fig. 1). Additionally, these systems can perform long dwell synthetic aperture radar scans for purposes of identification. A primary example of such a sensor is a multi-mode, electronically scanned antenna radar capable of tasking individual beams in terms of pointing direction, dwell time, and waveform. As illustrated in Fig. 1, an enhanced radar is capable of not only interleaving various radar beam modes (i.e., wide area search (WAS), sector search (SS), high range resolution (HRR), and synthetic aperture radar (SAR)), but will also be capable of scanning the surveillance region in an asynchronous fashion as the timeline suggests (e.g., irregular revisits could be due to sensor tasking to maintain tracks, search new areas, identify high-value targets, etc.). For such systems, the dynamic management of sensor mode control requires an automated process due to the variable timeline for adaptation.

Exploitation of sensor data from multi-mode sensors is capable of producing tactically significant information that can contribute to battlefield situation awareness. The multi-mode sensor data products contribute attributes of target detection, location, and classification together with environmental characteristics related to clutter. These attributes provide evidence necessary to produce a fused situation estimate. The challenge of sensor resource management for such systems is to characterize the exploitation and data production process according to a consistent model that provides for real-time adaptive sensor management.

In order to support the solution of the sensor management problem, several different solution approaches have been previously developed. They include information theoretic approaches [14], random set approaches [12], and the methods based on stochastic dynamic programming (SDP) [2–5]. The SDP algorithms were developed to address the problem of determining the optimal time sequencing of the radar's SAR (for detecting stationary objects) and MTI modes (for tracking moving objects) that maximizes the total *information value*. A *value function* is basically a function of low level tracking and classification quality states. The scheduler operates in a feedback manner in real time; for example, as objects are detected by the SAR, they may be eliminated from consideration by the scheduler so that the remaining radar resources can be better focused on only those objects remaining undetected or needing track improvement.

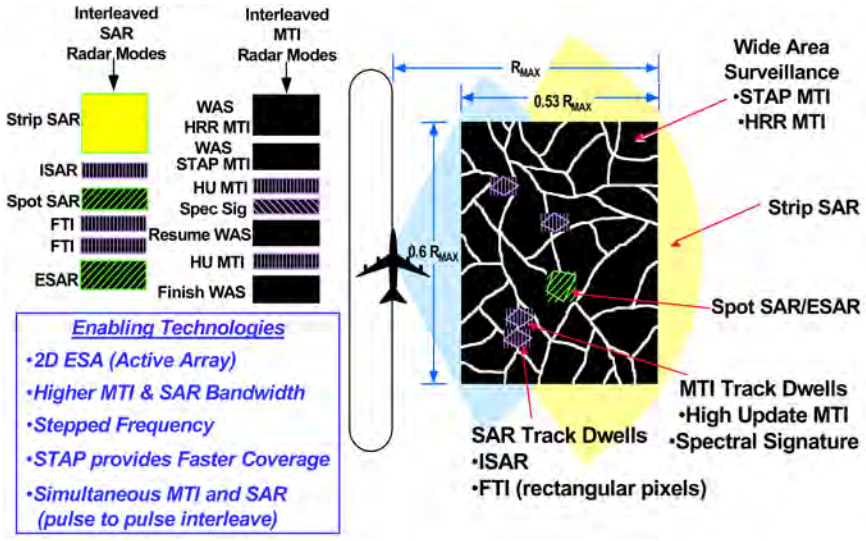


Fig. 1. Advanced airborne surveillance radars will include capabilities to operate in multiple modes and interleave modes.

The output of the scheduler is to determine for each instant of time whether the radar should

- operate in the SAR mode to image a single cell (for a stopped target), or
- operate in the MTI mode to detect and update tracks on all moving objects, or
- operate in the HRR mode to image a single cell (for a moving target) in order to obtain identification information (this is indeed the *only* way to obtain classification information), or,
- not be used at all in order to meet some exposure constraint.

The MTI mode typically requires the shortest dwell times (and thus consumes the least amount of radar resources); however, it has low range resolution and typically a low signal to noise ratio. The MTI mode is capable of detecting targets moving faster than the so-called minimum detectable velocity (MDV) of the sensor. It is well suited for problems such as tracking moving vehicles, characterizing traffic flow, and lines of communication using low-complexity (highest throughput) algorithms. The HRR mode has slightly longer dwell times (still shorter than SAR) but offers higher range resolution and strong signal to noise performance again for characterizing targets whose velocities exceed the MDV of the sensor. It has been proven to be useful for extracting coarse features (e.g. length and width) and as a tool for low-confidence classification. The HRR mode is eminently well suited for track maintenance problems. Finally, the SAR mode is ideal for two dimensional, high confidence classifications of stationary targets as well as change detection.

In the context of the Joint Director of Laboratories (JDL) terminology [17], it is observed that the track fusion models previously considered only address level 1 fusion (Object Assessment). Previous studies also indicated that sensor resource management algorithms utilizing only level 1 information may have limited value in certain operational scenarios involving multi-platform surveillance and strike missions because the response is optimized for track maintenance without any assessment of overall situation context [1, 13]. We contend that the problem of effective SRM for agile, multi-mode sensors will require improved representations of the process exploitation through level 2 information to adequately address the benefits of agile sensor tasking.

In this paper, we present a description of an algorithm that could be used to provide hierarchical target valuation based on not only level 1 (e.g., object or track) information, but also level 2 (e.g., groups of objects) information. This algorithm has the potential to improve the target valuation function used in a sensor resource manager by adding a valuation component related to the ability to identify a group of objects, such as convoys. This algorithm builds upon earlier results [10] that only addressed target valuation based on level 1 fusion information. The valuation algorithm is based on a Bayesian approach where a recursive composition inference algorithm was used to compute the hierarchical value function [10]. We have developed an efficient approximation algorithm to solve the combinatorial problem present in the original approach [7–8]. We have also developed an evaluation environment to analyze the performance of this valuation algorithm given a set of ground moving targets. The preliminary simulation results demonstrate the validity of our approach.

Note that the focus of this paper is on developing the hierarchical valuation function, not on deriving a sophisticated optimization algorithm. Essentially, any appropriate optimization algorithm can be applied to obtain the optimal solution. Although the stochastic dynamic programming approach mentioned before [3–5] seems to be a very suitable one. The remainder of the paper is organized as follows. Section 2 introduces and formulates the problem. Section 3 presents a complete

description of the valuation function and solution procedure. Section 4 describes the evaluation environment and the test scenarios followed by a set of simulation results given in Section 5 to demonstrate the new approach. Finally, our contribution and future research directions are summarized in Section 6.

2. PROBLEM DESCRIPTION AND SOLUTION CONCEPTS

Current Intelligence Surveillance and Reconnaissance (ISR) sensors can detect and take measurements on individual entities, such as moving vehicles and installations. These measurements can be used to infer the particular class of these individual entities. However, very few collection assets provide direct measurements on the hierarchical force structure of units that the entities comprise. Consequently, it is desirable to develop the capability to produce inferences on the hierarchical structure of military units based on inferences and measurements of individual entities and sub-units.

In many cases, the optimality of a sensor allocation policy is defined in terms of reduced tracking error and the best policy is determined through the solution of an optimization problem. While significant progress has been made in this area in the past, there remain open issues in the synthesis and validation of an approach to sensor resource management capable of utilizing high level fusion information.

A technique that can be used to assess the relative merit of aggregate force hypotheses from observations of a set of entities was presented in [10]. The technique draws inferences about the type of military unit that is present given partial observations of entities that comprise the units. Furthermore, making inferences about the type of military unit provides contextual information that enables improved inference about the type of individual vehicles. However, it was pointed out in [10] that the inference process involves intensive computations where the enumeration of an exponentially growing set is needed. In general, this could be very time consuming and may not be practical. In this paper, we develop an efficient approximate algorithm to resolve the combinatorial problem.

In a hierarchical data fusion functional model, high level processing includes estimation and prediction of relations among entities, force structure and cross force relations, communications and perceptual influences, physical context, etc. The goal of this paper is to develop models that estimate relations among entities which can contribute to force structure/composition assessment and to do this in a manner that enables the adaptive sensor management algorithms that have been previously developed.

Herein, we present a hierarchical value function (encompassing both level 1 and level 2 utility) using Bayesian Networks (BNs) [9] to implement sensor resource management algorithms. The valuation function includes both track level and higher-level (entity, convoy, group, scenario, etc.) information. Although significant research has been done in the area of sensor resource management as well as BNs with sensor fusion applications independently, to our knowledge, these two technologies have not been previously applied together to date to solve the higher-level fusion for sensor management problem.

3. SENSOR RESOURCE MANAGEMENT ALGORITHMS

As discussed earlier, the sensor has a capability of determining whether to collect MTI data at a dwell, or instead to collect HRR data on part of a dwell. The sensor management decisions will be based on information reported by a MHT (Multiple Hypothesis Tracker) [11, 15] which processes the sensor measurements.[1] This MHT also includes an ATR capability [6] which provides information on object estimated classifications based on the HRR measurements. The SRM algorithm can be considered as a controller which uses sensor actions to control the evolution of the information incorporated into the MHT algorithm. In order to make "good" sensor management decisions, it is important to model this evolution so that the SRM algorithm can predict the consequences of the alternative decisions.

However, the set of possible information states reported by the MHT for each track is very large. It consists of continuous variables with uncertainty (position, velocity, etc.), plus a set of probability distributions over target type, and discrete variables such as number of missed detection and status. Furthermore, the evolution of this information is highly uncertain, depending on the specific values of the future sensor measurements. For a large number of targets, the set of possible information states is the cross product of individual target states, leading to a large-dimensional continuous-valued state space. Designing feedback controls using such a state space would not lead to a practical real-time sensor management algorithm. An alternative approach is to characterize the information relevant to a track using an aggregate discrete-valued "information quality state" [1]. The model state has two components: tracking and classification quality. Each of these components can take a discrete number of values; thus, its evolution in response to sensor actions can be described by a finite-state Markov chain.

A. Track Quality State

In order to represent the behavior of the tracker algorithm, one way is to represent the track quality as a combination of tracking quality and classification quality. For example, in [1], the tracking quality state con-

[1]Note that the SRM algorithm could take information from any tracker, not necessary a MHT tracker.

sists of: *Undetected*, *Detected*, *New Track*, *Continuing Track*, *Coast 3 to go*, *Coast 2 to go*, *Coast 1 to go*,[2] and *Dropped*. The transition probabilities between these Markov states were given by parameters, which depend on the specific sensor mode being used by the radar, the sensor beam geometry for the track position indicating its probability of detection, and MHT parameters describing how tracks are nominated, promoted and dropped.

The target classification quality was modeled using a similar approach. For example, the classification information can be aggregated into four confidence levels: *Unclassified*, *Low Confidence*, *Medium Confidence*, and *High Confidence*. Note that the classification quality does not depend on the specific identity of an object; instead, it represents confidence in the identity assertion. Thus, the evolution model predicts the confidence improvement which results from specific sensor actions.

In [1], the transition probabilities of the two models were treated independently and computed as a function of the sensor mode separately. However, it is clear that the tracking and classification quality are correlated and should not be considered independently. We therefore develop a joint tracking and classification (JTC) quality state and develop a Markov model accordingly. By considering all the feasible combinations, the resulting model consists of 24 states as shown in Table I. The values (last column) assigned for each JTC state in Table I represent the relative preference of each state by the user. They are assigned heuristically and can be easily modified. More on the quality state value will be described in the next section. The Markov model transition diagram is shown in Fig. 2 and the corresponding transition matrix is given in Table II. Each entry in Table II represents the transition probability between two JTC states. Note that each row in the matrix needs to be normalized in order to make all the outgoing arcs from a state sum to 1.0. Also note that depending on the sensor mode, the transition matrix will be obtained based on the sensor parameters accordingly. The detailed description of the transition probabilities is given in Appendix A.

Given the representation of the information state described above, we can express the sensor management objectives as follows. First, we define the Tracking and Classification (TC) quality states and assign a numerical value $V(TC\_state)$ for every possible TC quality state. For example, a high numerical value would be assigned to a tracking quality of *Continuing_Track* and a classification quality of *High_Confidence*, whereas the lowest value would be assigned to a tracking quality of *Dropped_Track* (see Table I). It may be more important to track objects having a priority classification assessment, such as *time critical targets*. Thus we assume that there are values assigned to the different object classes. We then define an objective function which represents an overall tracking quality value given a sequence of sensor manager decisions. Note that the advantage of this method is that the aggregate Markov chain representation of information quality allows for fast prediction of MHT performance. The result is a practical, predictive model which can be used to evaluate trades between alternative sensor management decisions in real time.

TABLE I
24 State JTC Markov Model

Index	JTC	Tracking	Classification	Values
1	J11	Undetected	Unclassified	0
2	J21	Detection	Unclassified	1
3	J22	Detection	Low confidence	2
4	J31	False	Unclassified	0
5	J41	New track	Unclassified	2
6	J42	New track	Low confidence	3
7	J43	New track	Medium confidence	4
8	J51	Continuing track	Unclassified	7
9	J52	Continuing track	Low confidence	8
10	J53	Continuing track	Medium confidence	9
11	J54	Continuing track	High confidence	10
12	J61	Coast 3 to go	Unclassified	6
13	J62	Coast 3 to go	Low confidence	7
14	J63	Coast 3 to go	Medium confidence	8
15	J64	Coast 3 to go	High confidence	9
16	J71	Coast 2 to go	Unclassified	5
17	J72	Coast 2 to go	Low confidence	6
18	J73	Coast 2 to go	Medium confidence	7
19	J74	Coast 2 to go	High confidence	8
20	J81	Coast 1 to go	Unclassified	4
21	J82	Coast 1 to go	Low confidence	5
22	J83	Coast 1 to go	Medium confidence	6
23	J84	Coast 1 to go	High confidence	7
24	J91	Dropped track	Unclassified	0

B. Sensor Management Objectives

The SRM algorithm is based on an open-loop feedback approach. The basic idea is that at frame t, we generate the desired sequence of decisions for frames t, $t+1,\ldots,t+H$, where H is the planning horizon, based on the aggregate evolution represented by the information quality Markov chains in Fig. 2. We then collect the information from frame t, and receive updated track information from the MHT algorithm. Given this new information, we repeat the process and select decisions for frames $t+1,t+2,\ldots,t+H+1$, receive new information from the MHT and continue the iteration. Thus at each frame t, we compute sensor management decisions for several frames ahead, but use only the next frame's decisions to resolve the SRM problem. An important aspect of the sensor management methodology is that it decides immediate sensor mode commitments with a view towards how these decisions will affect the

[2]Coast 1 to go is the last tracking quality state before the track will be dropped.

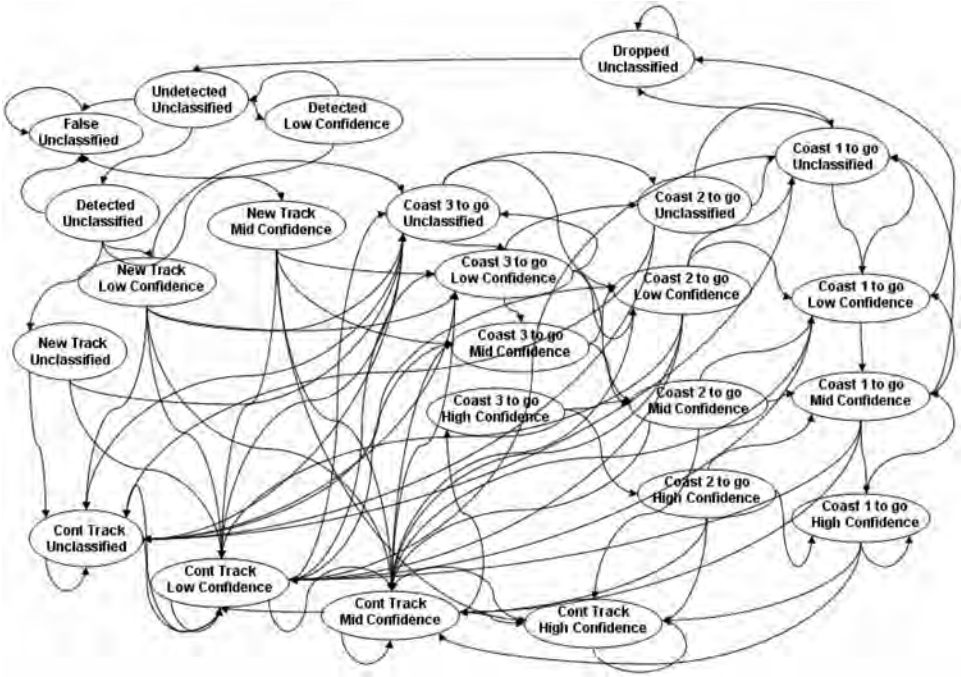


Fig. 2. Markov transition diagram of the joint tracking and classification quality state.

TABLE II
JTC Markov State Transition Matrix

JKC	J11	J21	J22	J31	J41	J42	J43	J51	J52	J53	J54	J61	J62	J63	J64	J71	J72	J73	J74	J81	J82	J83	J84	J91
J11	s1*s4	a1*s4	a1*a4	u1*s4																				
J21				u3*s4	a3*s4	a3*a4																		
J22				u3*u4	a3*u4	a3*s5	a3*a4																	
J31	0.1			0.9																				
J41								a3*s4	a3*a4			u3*s4												
J42								a3*u4	a3*s5	a3*a4		u3*u4	u3*s5											
J43									a3*u4	a3*s5	a3*a4		u3*u4	u3*s5										
J51								a2*s4	a2*u4			u2*s4												
J52								a2*u4	a2*s5	a2*a4		u2*u4	u2*s5											
J53									a2*u4	a2*s5	a2*a4		u2*u4	u2*s5										
J54										a2*u4	a2*s6			u2*u4	u2*s6									
J61								a2*s4	a2*u4			s2*s4	s2*u4			r2*s4								
J62								a2*u4	a2*s5	a2*a4		s2*u4	s2*s5	s2*a4		r2*u4	r2*s5							
J63									a2*u4	a2*s5	a2*a4		s2*u4	s2*s5	s2*a4		r2*u4	r2*s5						
J64										a2*u4	a2*s6			s2*u4	s2*s6			r2*u4	r2*s6					
J71								a2*s4	a2*u4							s2*s4	s2*u4			r2*s4				
J72								a2*u4	a2*s5	a2*a4						s2*u4	s2*s5	s2*a4		r2*u4	r2*s5			
J73									a2*u4	a2*s5	a2*a4						s2*u4	s2*s5	s2*a4		r2*u4	r2*s5		
J74										a2*u4	a2*s6							s2*u4	s2*s6			r2*u4	r2*s6	
J81								a2*s4	a2*u4											s2*s4	s2*u4			r2*s4
J82								a2*u4	a2*s5	a2*a4										s2*u4	s2*s5	s2*a4		r2*u4
J83									a2*u4	a2*s5	a2*a4										s2*u4	s2*s5	s2*a4	
J84										a2*u4	a2*s6											s2*u4	s2*s6	
J91	0.1																							0.9

information state H frames in the future. The size of H reflects a tradeoff between the desire to account for future actions versus the unpredictable evolution of future target motions. Larger values of H introduce more prediction uncertainty into future target positions, thereby making it harder to predict the effect of future sensor actions.

The SRM uses this information as follows. First, for each SRM track created from an MHT track, the classification probabilities of each track are used to

assign a value to this SRM track, as follows:

$$V(SRM\_Track) = \sum_{class \in classes} V(class) P(class \mid MHT\_Track) \tag{1}$$

where $class \in \{classes\}$ represents each of the possible target classification, $V(class)$ represents the decision maker's preference/priority on each target *class* and is assumed to be available.

The SRM objective for decisions selected at frame t is as follows. Given a set of SRM decisions for frames $t, t+1, \ldots, t+H$, for each SRM track, we can predict the probability distributions for the track quality state, $P_Q(\,)$, after the information from frame $t+H$ is processed. Denote these decisions as $u_t, u_{t+1}, \ldots, u_{t+H}$, the overall value of this sequence of decisions is computed as [1],

$$\begin{aligned} &J(u_t, u_{t+1}, \ldots, u_{t+H}) \\ &\quad = \sum_{\substack{SRM\_Track \\ \in SRM\_Tracks}} V(SRM\_Track) \\ &\qquad \times \sum_{\substack{JTC\_state \\ \in JTC\_states}} P_Q(JTC\_state \mid SRM\_Track) V(JTC\_state) \end{aligned} \tag{2}$$

where *SRM_Tracks* is the set of all *SRM_Track* and *JTC_States* is the set of all *JTC_State*. In (2), it is implicit that the track quality probabilities depend on the sequence of decisions. The SRM objective function described above represents an assignment of *value* to information quality and to classification of objects. This formulation couples the values of tracking and classification quality.

We will next show how to extend the target valuation models to include a hierarchical structure. The approach will be to modify the target valuation function (2) to include a higher level (cluster or unit) component. We rewrite (2) as

$$\begin{aligned} &J(u_t, u_{t+1}, \ldots, u_{t+H}) \\ &\quad = \sum_{\substack{SRM\_Cluster \\ \in SRM\_Clusters}} V(SRM\_Cluster) \left[\sum_{\substack{Cluster\_Track \\ \in Cluster\_Tracks}} V(Cluster\_Track) \sum_{\substack{JTC\_state \\ \in JTC\_states}} P(JTC\_state \mid Cluster\_Track) V(JTC\_state) \right] \end{aligned} \tag{3}$$

where

$$\begin{aligned} &V(SRM\_Cluster) \\ &\quad = \sum_{Unit\_types} V(Unit\_Type) P(Unit\_Type \mid Cluster\_Tracks) \end{aligned} \tag{4}$$

and *SMR_Cluster* is a group of tracks, denoted as *Cluster_Tracks*, linked together by proximity of a particular type of military unit.

As shown in (3), a *hierarchical target valuation model* is a function of both level 1 (Object Assessment) and level 2 (Situation Assessment) fusion quantities. The model is developed as follows:

1. First, we group the current MHT tracks into clusters.
2. For each cluster, we use a force structure model to infer unit type. We will construct a stochastic model by representing uncertainty (e.g. detection probability and unit composition variation).
3. We then develop a unit-level value function in addition to the entity level tracking and classification value functions as shown in (3). Note that the unit-level valuation function consists of two parts, $V(Unit\_Type)$ and $P(Unit\_Type \mid Cluster\_Tracks)$, as given in (4). $V(Unit\_Type)$ is the default value specified by the decision maker based on their preference/priority on each unit type and $P(Unit\_Type \mid Cluster\_Tracks)$ is the *Unit_Type* probability given a set of tracks computed by the BN force structure model to be discussed in the next section.

C. Bayesian Network Force Structure Model

With Bayes rule, the *Unit_Type* probability given a cluster of tracks can be computed by,

$$\begin{aligned} &P(Unit\_Type \mid Cluster\_Tracks) \\ &\quad = \frac{1}{c} P(Cluster\_Tracks \mid Unit\_Type) P(Unit\_Type). \end{aligned} \tag{5}$$

The solution to (5) represents one of the key contributions described in [10]. $P(Unit\_Type)$ represents the prior probabilities, and $P(Cluster\_Tracks \mid Unit\_Type)$ can be computed as follows:

$$\begin{aligned} &P(Cluster\_Tracks \mid Unit\_Type) \\ &\quad = \sum_{d \in D(n, r+1)} P(Cluster\_Tracks \mid d) P(d \mid Unit\_Type). \end{aligned} \tag{6}$$

In (6), $D(n, r+1)$ is the set of all possible distributions of the n detected vehicles into the $r+1$ possible vehicle classes (including the false-alarm class), which is a

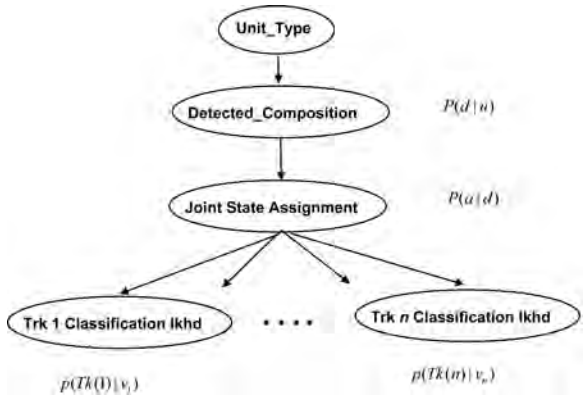


Fig. 3. A Bayesian network model for composition inference.

space with $(r+1)^n$ elements, $P(Cluster\_Tracks \mid d)$ is the likelihood of tracks classification states given the specific detection composition d, and $P(d \mid Unit\_Type)$ is the probability of detection composition given the unit type (see (8)). Note that in (6),

$$\begin{aligned} &p(Cluster\_Tracks \mid d) \\ &\quad = \sum_a p(Cluster\_Tracks \mid a) P(a \mid d) \\ &\quad = \sum_a \left[\prod_{k=1}^{n} p(Cluster\_Track(k) \mid v_k) \right] P(a \mid d) \end{aligned} \quad (7)$$

where in (7) $p(Cluster\_Track(k) \mid v_k)$ is the likelihood of producing a track classification state $Cluster\_Track(k)$ given a class v_k vehicle, and a is the joint assignment of a set of vehicles types to a set of tracks. Assuming all joint assignments consistent with the composition constraint are equally likely, then

$$P(a \mid d) = \begin{cases} 1/|\Omega(d)|, & a \in \Omega(d) \\ 0, & \text{otherwise} \end{cases}$$

where

$$|\Omega(d)| = C^{n(d)}_{n(1;d),\ldots,n(r;d)} = \frac{[n(1;d) + \cdots + n(r;d)]!}{n(1;d)! \cdots n(r;d)!}$$

is the set of all joint assignments in which $n(v;d)$ is the number of detected class v vehicles in d.

Also in (6), from the detection model (for simplicity, $u \equiv Unit\_Type$ will be used in the following equations),

$$P(d \mid u) = p_o(n(0;d); \lambda_{FA}) \prod_{v=1}^{r} P(n(v;d) \mid n(v;u)). \quad (8)$$

In (8), $n(0;d)$ is the number of false detections, $n(v;u)$ is the number of class v vehicles in a type u unit,[3] $p_o(k;\lambda) = \lambda^k e^{-\lambda}/k!$ is the Poisson distribution for false alarm detection probability, and

$$\begin{aligned} &P(n(v;d) \mid n(v;u)) \\ &\quad = \sum_{k=0}^{\min[n(v;u), n(v;d)]} B(k; n(v;u), P_D(v)) \cdot p_o(n(v;d) - k; \lambda_C(v)) \end{aligned} \quad (9)$$

[3]Note that the composition of each unit type is assumed to be given.

is the probability of target detection with $B(k;n,p) = C^n_k p^k (1-p)^{n-k}$, a Binomial distribution, where $\lambda_C(v)$ is the density of confusers of class v vehicle.

To implement the hierarchical valuation function, one way is to use the BN model constructed based on (6)–(9) as shown in Fig. 3. Many efficient algorithms exist for BN probabilistic inference [16]. However, (6)–(9) involve intensive computations where the enumeration of an exponentially growing set is needed. In general, this could be very time consuming and may not be practical. We have thus developed an approximate method to simplify the approach, namely,

$$\begin{aligned} &P(Cluster\_Tracks \mid Unit\_Type) \\ &\quad \approx P(d(Cluster\_Tracks) \mid Unit\_Type) \end{aligned} \quad (10)$$

where $d(Cluster\_Tracks)$ is defined as the joint detection-classification state by collapsing all the track classification probability distributions into one. Namely,

$$d(Cluster\_Tracks) = \sum_{\substack{Cluster\_Track \\ \in Cluster\_Tracks}} P_C(Cluster\_Track),$$

where $P_C(Cluster\_Track)$ is the classification probability distribution of the track $Cluster\_Track$. Note that $d(Cluster\_Tracks)$ is the expected number of targets of each class. Essentially, the approximation amounts to replacing the distribution over numbers of each vehicle type by the mean number of each vehicle type. Then

$$\begin{aligned} &P(d(Cluster\_Tracks) \mid Unit\_Type) \\ &\quad = \prod_{v=1}^{r} P_B(n(v; d(Cluster\_Tracks)) \mid n(v;u)) \end{aligned} \quad (11)$$

where $P_B(n(v;d(Cluster\_Tracks)) \mid n(v;u))$ is defined similarly to (9). However, since $d(Cluster\_Tracks)$ is a vector of positive real numbers (not necessary integers), it may not be possible to perform the calculation in (9). We therefore approximate it by

$$P_B(n(v;d(Tks)) \mid n(v;u)) \approx N(n(v;d(Tks)); \bar{n}, \sigma_n^2) \quad (12)$$

where $N(n;\bar{n},\sigma_n^2)$ is a Gaussian distribution, $\bar{n} = n(v;u) \cdot P_D(v) + \lambda_C(v)$ is the expected number of detected class v targets, and $\sigma_n^2 = \max[\sigma_{\min}, n(v;u) P_D(v)(1 - P_D(v)) + \lambda_C(v)]$ is the approximate associated variance. With (10)–(12), the composition inference becomes significantly simpler and much more efficient to compute.

4. SENSOR RESOURCE MANAGEMENT EVALUATION ENVIRONMENT

In order to test our target valuation algorithms, we implemented a simple test system as shown in Fig. 4. Note that the purpose of this architecture was not to provide high fidelity modeling conditions; rather, it was designed to be quickly constructed for the purpose of evaluating the proof-of-concept target valuation algorithm(s) that were developed in this paper.

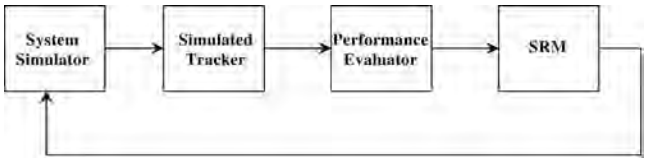


Fig. 4. A SRM system architecture.

The architecture contains an outer loop where at each instant of time, the system simulator creates and sends the current scenario information (related to the targets and sensors) to the simulated tracker. The simulated tracker then produces the simulated tracks with proper joint (tracking and classification) quality states and classification vectors based the Markov transition model as well as the true target/sensor parameters. The simulated tracker then sends the track results to the evaluator. The evaluator uses the tracking results to determine the best sensor mode and pointing direction for next instant of sampling time and sends that decision back to the system simulator. In the remainder of this section, we will provide a description of each of the components in Fig. 4.

A. System Simulator

The system simulator is the overall driver of the system. It generates the ground truth scenarios including target trajectories, group/convoy composition, sensor placements, and sensor observations based on sensor mode/characteristics, as well as sensor/target geometry. Note that the two important aspects of the simulator are: (a) the ability to simulate group/convoy behavior, and (b) the ability to switch sensor operating modes based on the information supplied by the Performance Evaluator and SRM components.

The system simulator sends parameters such as the number of targets, their locations and classes, group composition/identity, sensor mode, detection probability, false alarm density, target density, and confusion matrix/classification probability to the Simulated Tracker. The relative target/sensor geometry (which accounts for a target dropping below the sensor's Minimum Detectable Velocity (MDV) for MTI) is incorporated by the system simulator to produce the required operating parameters.

For simplicity, we leave the burden of representing the convoys/units to the system simulator. Namely, the system simulator will send both target and convoy/group information to the simulated tracker. In the simulation, we assume coverage of all targets in the test scenario area of interest (AOI) when the sensor is in the MTI mode. On the other hand, the HRR mode has a more limited FOV depending on the sensor pointing direction.

B. Simulated Tracker

The purpose of this module is to produce simulated tracking results for performance evaluation without implementing a real tracker. The simulated tracker receives inputs such as ground truth, sensor models, etc. from the system simulator and produce a set of tracks each with tracking and classification joint quality state. Note that for every ground truth target, there is a "track" which will be in one of the joint quality states at each instance of time. For test and validation purposes, we did not attempt to use a complete Multiple Hypothesis Tracker (MHT) for tracking moving ground targets. Rather, the simulated tracker was designed to estimate the joint tracking and classification quality state for each target track.

As described in Section 3, the track quality states behave according to a Finite State Markov transition model based on sensor mode and operating conditions. There are three choices available to the SRM: to use the MTI mode, to use the HRR mode, or to not use the sensor at all. Each sensor mode represents an action that the sensor can take to observe targets. Note that this is a simplified form of the state transitions that were presented earlier, which were chosen on the basis of their ease of implementation.

In addition to the joint quality states, in order to evaluate the track valuation function, each track needs to have an *a posteriori* classification probability distribution. The simulated tracker produces the classification probability vector based on the true target class, previous track class probability, and the current sensor mode and operating conditions. For example, at the beginning of the simulation, each track is at untracked/unclassified state with a uniform classification probability. Depending on the sensor mode (of the next sampling time) and operating characteristics, the track quality state at the next sampling time will be simulated stochastically based on a Finite State Markov transition model. The track classification probability is also updated (accrued over time) using the confusion matrix of the particular sensor mode and the simulated sensor observations. For example, the confusion matrix for GMTI mode is a matrix consisting of uniform probabilities since GMTI mode has no ability to classify target. On the other hand, each row of the HRR mode confusion matrix is the probability of observed classification given a true target class. Note that for the kinematic state, the simulated tracker does not represent the tracks in the target state space, but rather only on the quality state space. The simulated tracker then sends the track results to the evaluator.

C. Performance Evaluator

As mentioned previously, larger values of H (the planning horizon) introduce more prediction uncertainty into future target positions, thereby making it harder to predict the effect of future sensor actions. However, for test purpose we assumed that $H = 1$, (i.e., a one-step look ahead) in order to simplify the implementation (this avoids the need for implementing a dynamic programming algorithm) and to focus on validating the higher level valuation algorithm.

The evaluator receives track results from the simulated tracker. In the results, there is a set of clusters/groups/convoys where each group contains a set of tracks. As described before, each track has a joint quality state and a classification probability vector. The evaluator will use the track results to determine the best sensor mode and pointing direction for the next detection time. In order to do so, in addition to the information from the tracker, truth-related domain knowledge such as unit composition, decision maker's preference value (for each target class and convoy/unit type) are needed as well. The evaluator first finds all the possible detected compositions (based on the possible unit types) and map these to the track compositions of each cluster, which are produced by the simulated tracker. It then computes the value function based on a hierarchical Bayesian model/inference (as described in Section 3) which takes into account the track classification likelihood of a joint state assignment. The evaluator then produces a best sensor mode decision (HRR or MTI) and pointing direction based on the resulting information value and sends these decisions back to the SRM to be used by the system simulator for the next sampling time.

Note that to evaluate the overall system performance, the probability distributions of each unit and individual target are produced by the simulated tracker based on the selected sensor decision. The tracking results are then averaged over multiple Monte Carlo runs as will be described in the next section.

5. SIMULATION RESULTS

We implemented the system described in Fig. 4 and also defined a set of metrics that can be used for evaluation purposes:

- *Sensor allocated resources*—the percentage of time that the sensor is operating in HRR mode (HRR Rate)
- *Average probability of correct unit classification*—the average correct unit classification probability over the simulation time (P_{cc})
- *Average percentage of correctly identified targets*—the average percentage of correct target classification in each unit over the simulation time (Trk Rate)

In order to test these algorithms, we implemented a simple ground moving target scenario containing convoy units, each consisting of a different combination of target types. There are a total of 4 possible types of unit: Scud (class 1), C2 (class 2), Tank (class 3), and Unknown, as well as 6 target classes: UAZ-469 (class 1), ZIL-151 (class 2), GAZ-66 (class 3), MAZ-543 (class 4), T-72 (class 5), and Other (class 6). However, in this particular scenario, ground truth only contains two units and four target classes: unit 1 (containing 2 UAZ-469 vehicles, as well as one of ZIL-151, GAZ-66, and MAZ-543 each) and unit 2 (containing 5 UAZ-469 vehicles).

The parameters used for the simulation are summarized in Appendix B. We made several test trials with different value functions and strategies. Specifically, in each of the combinations below, the notation "Case xy" refers to x = *unit_type* and y = *target_class*. Also, the notation [a b c d] refers to the valuation of each unit (since there are 4 possible units for this particular scenario) and the notation [a b c d e f g] similarly refers to the valuation of individual targets, since there are 6 possible targets. Note that for simplicity, the target valuation is assumed to be a "binary" variable (i.e., valued at either 0 or 1); other combinations of target values are certainly possible, but were not considered here. This means that, for this particular scenario, there are a total of 15 possible combinations, as follows: 2 (*unit classes*) $*$ 4 (*target classes*) + 4 (*level 1 valuation only*) + 2 (*level 2 valuation only*) + 1 (*neither level 1 nor level 2 valuation*) = 15. Also, we assume the cost of using the HRR mode is about twice as expensive than the MTI mode.

Level 1 value function only, unit class value [1 1 1 1]
- Case 01: Focus on target class 1, track class value: [1 0 0 0 0 0]
- Case 02: Focus on target class 2, track class value: [0 1 0 0 0 0]
- Case 03: Focus on target class 3, track class value: [0 0 1 0 0 0]
- Case 04: Focus on target class 4, track class value: [0 0 0 1 0 0]

Level 2 value function only, target class value [1 1 1 1 1 1]
- Case 10: Focus on unit class 1, unit class value: [1 0 0 0]
- Case 20: Focus on unit class 2, unit class value: [0 1 0 0]

Level 1 and 2 value functions
- Case 11: Focus on unit class 1, [1 0 0 0], and target class 1, [1 0 0 0 0 0]
- Case 12: Focus on unit class 1, [1 0 0 0], and target class 2, [0 1 0 0 0 0]
- Case 13: Focus on unit class 1, [1 0 0 0], and target class 3, [0 0 1 0 0 0]
- Case 14: Focus on unit class 1, [1 0 0 0], and target class 4, [0 0 0 1 0 0]
- Case 21: Focus on unit class 2, [0 1 0 0], and target class 1, [1 0 0 0 0 0]
- Case 22: Focus on unit class 2, [0 1 0 0], and target class 2, [0 1 0 0 0 0]
- Case 23: Focus on unit class 2, [0 1 0 0], and target class 3, [0 0 1 0 0 0]
- Case 24: Focus on unit class 2, [0 1 0 0], and target class 4, [0 0 0 1 0 0]

Neither level 1 nor level 2 value functions
- Case 00: unit class value, [1 1 1 1], and target class value, [1 1 1 1 1 1]

Note that for Case 00, in order to ignore target (or unit) value, all of the entities are equally valued and have a value of 1, e.g., a target class of [1 1 1 1 1 1].

TABLE III
Performance Results using Only Level 1 Value Function

Case	HRR U1 Rate	HRR U2 Rate	U1 Avg P_{cc}	U2 Avg P_{cc}	U1 Trk Rate	U2 Trk Rate
00	0.0032	0.0020	0.5671	0.5466	0.3173	0.3140
01	0.0000	0.0260	0.5000	0.8596	0.2500	0.8585
02	0.0036	0.0000	0.5838	0.5000	0.3405	0.2500
03	0.0068	0.0000	0.6502	0.5000	0.4242	0.2500
04	0.0076	0.0000	0.6656	0.5000	0.4243	0.2500

TABLE IV
Performance Results using Only Level 2 Value Functions

Case	HRR U1 Rate	HRR U2 Rate	U1 Avg P_{cc}	U2 Avg P_{cc}	U1 Trk Rate	U2 Trk Rate
00	0.0032	0.0020	0.5671	0.5466	0.3173	0.3140
10	0.0236	0.0000	0.9746	0.5000	0.8150	0.2500
20	0.0000	0.0236	0.5000	0.8925	0.2500	0.7777

TABLE V
Performance Results using Both Level 1 and Level 2 Value Functions

Case	HRR U1 Rate	HRR U2 Rate	U1 Avg P_{cc}	U2 Avg P_{cc}	U1 Trk Rate	U2 Trk Rate
10	0.0236	0.0000	0.9746	0.5000	0.8150	0.2500
11	0.0240	0.0000	0.9868	0.5000	0.8162	0.2500
12	0.0224	0.0000	0.9127	0.5000	0.7482	0.2500
13	0.0212	0.0000	0.9181	0.5000	0.7790	0.2500
14	0.0244	0.0000	0.9695	0.5000	0.8148	0.2500
20	0.0000	0.0236	0.5000	0.8925	0.2500	0.7777
21	0.0000	0.0336	0.5000	0.9288	0.2500	0.9088
22	0.0008	0.0056	0.5182	0.6089	0.2721	0.3922
23	0.0000	0.0048	0.5000	0.5730	0.2500	0.3586
24	0.0004	0.0068	0.5000	0.6219	0.2500	0.4301

With 50 Monte Carlo simulations, the average performance for several different cases are summarized in the following tables. Table III shows the performance results using only level 1 valuation functions. Note that case 00, by definition, contains neither level 1 nor level 2 valuation functions. In the table, the HRR rates (percentage of time spent in HRR mode versus GMTI mode) for unit 1 and unit 2 are shown in columns 2 and 3, average correct classification probabilities for unit 1 and unit 2 are given in cloumns 4 and 5, and the average correct track classification probabilities of each unit are shown in the last two cloumns respectively.

The results show that the classification performance of both units are in the range of 50–60%. Both unit 1 and unit 2 classifications improve somewhat when adding the "right" level 1 valuation. For example, when adding target class 2, 3, and 4 valuation functions, unit 1 classification increases from 50 + % to around 60 + % while unit 2 classification drop slightly from 55% to 50%. Similarly, when adding a target class 1 valuation function, unit 1 classification decreases to 50% while unit 2 classification increases significantly to 86%. This is understandable since unit 1 consists of all 4 classes of targets and unit 2 contains only class 1 targets.

Table IV shows the corresponding performance results using only level 2 value functions. It can be seen that the classification performance improves significantly compared to the level 1 performance. For example, for case 10, since the emphasis (i.e., valuation) is on unit 1, the resulting sensor strategy improves the unit 1 classification performance significantly from 57% to 97%. Similarly, for case 20, unit 2 classification increases from 55% to 89%. Note here that the track level classification probabilities also improve significantly from around 30% to 80% despite no level 1 valuation being used.

Table V shows the performance results using both level 1 and level 2 value functions. It can be seen that the classification performance values are similar to those obtained using only level 2 value functions. For example, in cases 11 through 14, since the emphasis is on unit 1, the unit 1 classification performance is similar to that of case 10. However, for cases 21 through 24, only case 21 performs similarly to case 20—the others perform significantly worse in being able to classify unit 2. This is because, interestingly enough, unit 2 includes only target class 1. Adding a level 1 track value function of the other target class (not included in unit 2) not only does not help in classifying unit 2, but it also manages to confuse the sensor manager and subsequently deteriorates the performance (relative to case 20) significantly.

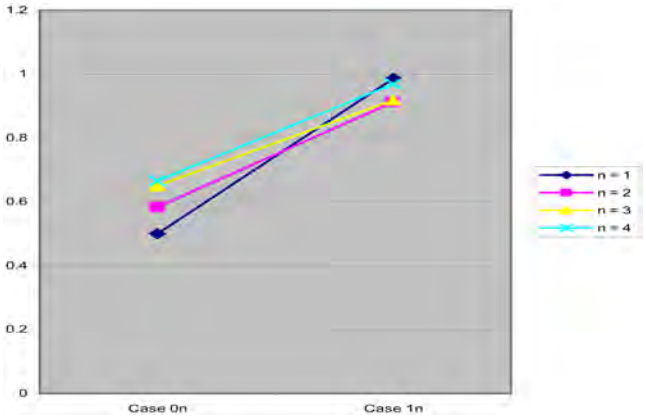


Fig. 5. Unit 1 average Pcc with level 1 and level 2 valuation functions.

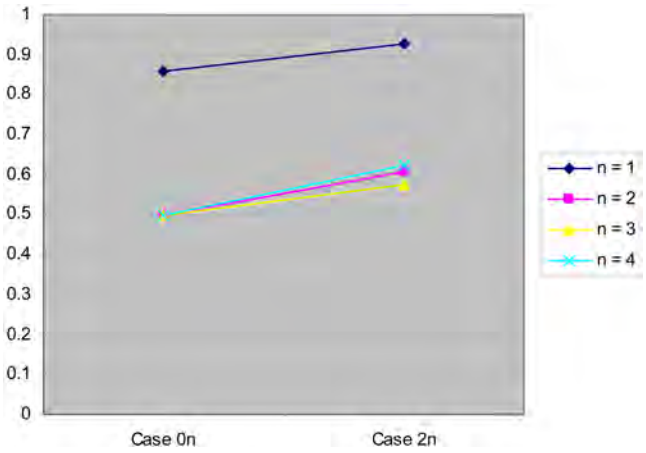


Fig. 6. Unit 2 average Pcc with level 1 and level 2 valuation functions.

Another useful comparison is to determine the possible benefit of adding a level 2 valuation function to the level 1 function. This comparison is particularly relevant because use of level 1 (target-based) valuation can be considered to be the typical baseline or current operating mode for most sensor management systems. In order to perform this comparison, we extract comparable portions of Table III and Table V and display these values in Figs. 5 and 6, where the comparison is between case $0n$ with cases $1n$ and $2n$ respectively, where $n = 1,\ldots,4$. Note that case $0n$ represents level 1 valuation that emphasizes target class n and case jn represents level 1 and level 2 valuations that emphasize unit type j and target class n.

As seen from the figures, in all 4 cases, unit 1 P_{cc} improve significantly (from about 60% to 90%) when adding level 2 valuation function. However, the unit 2 P_{cc} only improve moderately (about 10%) when adding a level 2 function that emphasizes unit 2 valuation. Again, this is because unit 2 consists of only target type 1, the combination of "inconsistent" unit level and target level values (such as 22, 23, and 24) simply will not help the classification performance.

In summary, as seen in the results, by adding a level 2 valuation function, the performance improves significantly. Particularly, without level 2 function, the performance for unit level classification is mostly unsatisfactory. It is interesting to note that, with a level 2 valuation function, the track level performance also improve slightly when the objective functions of the two levels are consistent. When the objective functions are inconsistent or contradictory as we described above, the performance may not improve as expected. Note that since we are not comparing performance between the proposed approach and an alternative baseline, the numerical results imply, not so much that performance uniformly improves when optimizing the proposed objective criterion but that the objective criterion is a reasonable one to optimize to meet both level 1 and 2 fusion objectives.

6. SUMMARY

In this paper, we have presented an approach for dynamically choosing sensor mode and pointing direction based on both level 1 and level 2 information. Specifically, a hierarchical target valuation model based on track quality value was presented. The valuation algorithm relies on a Bayesian approach where a recursive composition inference algorithm was used to compute the hierarchical valuation function. This approach not only will provide for adequate object identification and tracking performance, but also can provide the ability to be able to identify higher-level entities such as convoys.

We have also developed an evaluation environment to analyze the performance of this valuation algorithm given a set of ground moving targets. The preliminary simulation results demonstrate the validity of our approach. In order to completely validate algorithm performance, it will be necessary to implement the algorithm in a higher fidelity modeling environment, including more complex algorithms for the tracker and SRM. Nevertheless, the algorithms presented in this paper represent a significant step toward efficient sensor management using higher level valuation and objective functions. Some useful future research directions include extending the hierarchical valuation model to account for level 3 (eg., intent assessment) function and developing a analytical prediction model to estimate the SRM performance without extensive Monte Carlo simulations.

APPENDIX A

The parameters in the transition matrix of the tracking and classification quality Markov chains are defined as the follows.

(1) $a_1 = P_{new}P_d$ (object, sensor_mode), $u_1 = 1 - a_1$, $s_1 = 0$, when a potential track is covered by the sensor

beam, where P_{new} is the probability of new target arrivals per unit position, and P_d is the probability of detection calculated based on the relative predicted target-sensor geometry and the sensor mode as well as the target radial velocity (for MDV purposes).

(2) $u_1 = a_1 = 0$ and $s_1 = 1$ when a potential track is not covered by the sensor.

(3) $a_3 = P_d$ (*object, sensor_mode*) is the probability that a second beam look results in an initial track, $u_3 = 1 - a_3$.

(4) $a_2 = P_d$ (*object, sensor_mode*), $r_2 = 1 - a_2$, $s_2 = 0$, when the object is covered by the sensor beam.

(5) When the object is unobservable, $a_2 = 0$, $r_2 = rate$, $s_2 = 1 - r_2$, where $rate = 3F_d/T_{drop}$ is the probability an unobservable object will reach the dropped state in T_{drop} expected time. Note that F_d is the frame duration and T_{drop} is the maximum time that a track can be kept coasting before the MHT algorithm drops the track.

(6) $a_4\ (HRR) = P_{improve}$, $a_4\ (MTI) = 0$, $s_4 = 1 - a_4$ where $P_{improve}$ is the probability that classification quality will improve if an HRR model is used.

(7) $u_4\ (MTI) = P_{degrade}$, $u_4\ (HRR) = 0$, $s_5 = 1 - a_4 - u_4$, $s_6 = 1 - u_4$ where $P_{degrade}$ is the probability that classification quality will degrade if an MTI model is used.

APPENDIX B

The parameters in the simulation are given as the follows.

(1) probability of detection: $P_d\ (GMTI) = 0.9$, $P_d\ (HRR) = 0.5$

(2) probability of classification: $P_C\ (GMTI) = 1/n$, $P_C\ (HRR) = 0.9$[4]

(3) probability of new target arrivals per unit position: $P_{new} = 0.1$

(4) the probability that classification quality will improve with HRR mode: $P_{improve} = 0.8$

(5) the probability that classification quality will degrade with GMTI mode: $P_{degrade} = 0.8$

(6) the probability an unobservable object will reach the dropped state: $rate = 0.1$

(7) the values of joint quality state: given in the last column of Table I.

(8) Decision maker's preference value for each tareget and unit: varied in each test case, see Section 5.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers and Editor for their constructive comments that helped improve the paper.

[4]It is assumed that the probability of misclassification is uniform across target types.

REFERENCES

[1] T. Allen, D. Castañon and R. Washburn
Stochastic dynamic programming for farsighted sensor management.
Phase II SBIR Final Report, ALPHATECH Technical Report, TR-974, June 2000.

[2] R. Popp, A. Bailey and J. Tsitsiklis
Dynamic airborne sensor resource management for ground moving target tracking and classification.
In *Proceedings of IEEE Aerospace Conference*, Big Sky, MO, Mar. 2000.

[3] D. Bertsekas and D. Castañon
Rollout algorithms for stochastic scheduling.
Heuristics, **5** (1999).

[4] D. Bertsekas and J. Tsitsiklis
Dynamic Programming and Optimal Control vol. I, II.
Belmont MA: Athena Scientific, 2001.

[5] D. Castañon
Approximate dynamic programming for sensor management.
In *Proceedings of the 1997 Conference on Decision and Control*.

[6] K. Chang and R. Fung
Target identification with Bayesian networks in a multiple hypothesis tracking system.
SPIE Optical Engineering Journal, **36**, 3 (Mar. 1997), 684–691.

[7] J. Hill and K. Chang
Improved representation of sensor exploitation for automatic sensor management.
In *Proceedings of Sixth International Conference on Information Fusion*, Cairns, Queensland, Australia, July 2003, 688–694.

[8] J. Hill and K. Chang
Sensor resource management with hierarchical target valuation models.
In *Proceedings of Seventh International Conference on Information Fusion*, Stockholm, Sweden, July 2004.

[9] F. Jensen
An Introduction to Bayesian Networks.
Spinger, NY, 1996.

[10] J. Johnson and R. Chaney
Recursive composition inference for force aggregation.
In *Proceedings of Second International Conference on Information Fusion*, July 1999.

[11] T. Kurien
Issues in the design of practical multitarget tracking algorithms.
In Y. Bar-Shalom (Ed.), *Multitarget-Multisensor Tracking: Applications*, Artech House, 1990, ch. 2.

[12] R. Mahler
Random set theory for target tracking and identification.
In D. L. Hall and J. Llinas (Eds.), *Handbook of Multisensor Data Fusion*, Boca Raton FL: CRC Press, 2002, ch. 14.

[13] R. Mahler
Target preference in multitarget sensor management: A unified approach.
In *Proceedings of SPIE conference*, Orlando, FL, 2004.

[14] J. Manyika
An information-theoretic approach to data fusion and sensor management.
Ph.D. thesis, University of Oxford, 1993.

[15] S. Mori, C. Y. Chong, E. Tse and R. Wishner
Tracking and classifying multiple targets without a priori identification.
IEEE Transaction on Automatic Control, **AC-31**, 5 (1985), 401–409.

[16] J. Pearl
Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.
Morgan Kaufmann, 1989.

[17] A. Steinberg, C. Bowman and F. White
Revisions to the JDL data fusion model.
In *Proceedings of SPIE, Sensor Fusion: Architectures, Algorithms, and Applications III*, 1999.

[18] R. B. Washburn, M. K. Schneider and J. J. Fox
Stochastic dynamic programming based approaches to sensor resource management.
In *Proceedings of the Fifth International Conference on Information Fusion*, Annapolis, MD, July 2002.

K. C. Chang received the M.S. and Ph.D. degrees in electrical engineering from the University of Connecticut in 1983 and 1986 respectively.

From 1983 to 1992, he was a senior research scientist in the Advanced Decision Systems (ADS) division, Booz-Allen & Hamilton, California. In 1992, he joined the Systems Engineering and Operations Research Department, George Mason University where he is currently a professor. His research interests include estimation theory, optimization, signal processing, and multisensor data fusion. He is particularly interested in applying unconventional techniques to the conventional decision and control systems. He has more than 25 years of industrial and academic experience and has published more than one hundred and forty papers in the areas of multitarget tracking, distributed sensor fusion, and Bayesian Network technologies. He was the editor on Large Scale Systems of *IEEE Transactions on Aerospace and Electronic Systems* and an associate editor of *IEEE Transactions on Systems, Man, and Cybernetics.*

Dr. Chang is a member of Etta Kappa Nu and Tau Beta Pi.

Joe Hill received the M.S. degree in electrical engineering (Systems and Applied Mathematics) from the University of Notre Dame in 1982 and his B.S.E.E. degree from Valparaiso University in 1979, with highest distinction.

He has developed error covariance models for Submarine Launched Ballistic Missile weapon and Navigation systems (TASC, 1984–1991), developed hit to kill models of missile/interceptor endgame performance in support of the Missile Defense Agency (Coleman Research, 1991–1996), and developed and tested collection performance/geolocation models for several overhead sensor systems (Ultrasystems/Logicon/Northrop Grumman Information Technology, 1996–2001). From 2001 to 2007, he was a senior engineer with ALPHATECH, Inc. (now a part of BAE Systems). He is currently a senior engineering specialist with SRS Technologies. His current areas of interest are sensor modeling, scheduling, tracking performance, route planning, resource management, control, and optimization.

He is a member of Tau Beta Pi, and ISIF.

An Information Fusion Game Component

JOEL BRYNIELSSON
Swedish National Defence College

STEFAN ARNBORG
Royal Institute of Technology

Higher levels of the data fusion process call for prediction and awareness of the development of a situation. Since the situations handled by command and control systems develop by actions performed by opposing agents, pure probabilistic or evidential techniques are not fully sufficient tools for prediction. Game-theoretic tools can give an improved appreciation of the real uncertainty in this prediction task, and also be a tool in the planning process. Based on a combination of graphical inference models and game theory, we propose a decision support tool architecture for command and control situation awareness enhancements.

This paper outlines a framework for command and control decision-making in multi-agent settings. Decision-makers represent beliefs over models incorporating other decision-makers and the state of the environment. When combined, the decision-makers' equilibrium strategies of the game can be inserted into a representation of the state of the environment to achieve a joint probability distribution for the whole situation in the form of a Bayesian network representation.

Manuscript received December 31, 2004; revised January 30, 2006 and June 11, 2006; released for publcation on July 30, 2006.

Refereeing of this contribution was handled by Shozo Mori.

Authors' addresses: J. Brynielsson, Swedish National Defence College, PO Box 27805, SE-115 93 Stockholm, Sweden, E-mail: (joel@kth.se); S. Arnborg, Royal Institute of Technology, SE-100 44 Stockholm, Sweden, E-mail: (stefan@nada.kth.se).

1. INTRODUCTION

The military domain is one of the purest possible game arenas, and history is full of examples of how mistakes in handling uncertainty about the opponent have had large consequences. For an entertaining selection, see, e.g., [16]. Commanders on each side have resources at their disposal, and want to use them to achieve their, mostly opposing, goals. In the network centric warfare [1, 2] era, they are aided by large amounts of information about the opponent from sensors and historical data bases, and about the status of their own resources from their own information technology infrastructure. In recently proposed infostructures for command and control (C2) [12], decision support tools play a prominent role. These tools seldom include game-theoretic means. Gaming is, however, a prominent feature of military training and the regulated decision processes often assign the roles of red and blue players to staff officers in manual planning activities [52]. Gaming is thus a conceptual part of the planning process in many organizations. It must be emphasized, however, that there are significant differences between practice and theory in application of such regulations. It has, for example, been shown in studies that the Swedish defense organization practices a more naturalistic decision-making process than the recommended one [51]. A pure naturalistic planning process relies more on unobservable mental capabilities of decision-makers than on rational analyses of alternative moves and their utilities [28]. The most common way to deal with uncertainty is, however, to make an assumption—and to forget that it was made. These observations have been the starting point for introducing a less complex planning model—PUT (Planning Under Time-pressure)—in the Swedish defense organization. PUT is based on analyses of a few opponent alternatives and incremental improvement of one's own plans [51]. It thus has potential for the use of gaming tools, provided they are realized in a way that supports subjective improvement of decision situations and decision quality [3].

Data fusion aims at providing situation awareness at different levels for a commander. The JDL model [47, 56] has been proposed for structuring the fusion process into five levels where the third level consists of higher level prediction of possible future problems and possibilities. We believe that the problem of predicting the future in a C2 context comes in two variations that differ in complexity and dependencies: the problem of capturing all aspects of a complex situation, and the problem of strategic dependence in a multi-agent encounter. Considering the former problem, the influence diagram is a well-established and appropriate modeling technique for modeling everything that is not dependent on our own or the opponents' actions, for example doctrine and terrain. Efforts in this direction have been proposed; see for example [50] for a discussion about doctrine modeling using dynamic Bayesian networks [40].

Looking at the latter problem, predicting decisions is also a game-theoretic problem, which has been noted in a recent proposal for revisions to the JDL model where the authors suggest the use of game-theoretic algorithms for the estimation process in higher level data fusion [37].

In this paper, we outline a schematic model using influence diagrams to obtain parameters for a description of the situation in the form of a Bayesian game. The result from the game is a description of equilibrium strategies for participants that can be incorporated in the influence diagram to form a Bayesian network (BN) description of the situation and its development, changing decision nodes to chance nodes.

We review some applications of gaming and simulation in Section 2 and describe our use of influence diagrams in Section 3. Section 4 gives background and some historical notes regarding agent interaction and Section 5 gives a short background on game theory. Section 6 contains an outline of the game component representation. Section 7 discusses solutions and addresses the problem of obtaining these solutions in a computationally feasible manner. In Section 8 we illustrate the use of Bayesian game-theoretic reasoning for operations planning by transforming a decision situation into a Bayesian game that we solve. Section 9 addresses the problems and possibilities that the ambiguities typical for a game-theoretic solution pose. Section 10 discusses related work and Section 11 is devoted to conclusions and discussion regarding future research.

2. THE GAMING PERSPECTIVE

Tools proposed to support the gaming perspective include microworlds [10, 17, 35], which are computer tools where several operators train together; and computer-intensive sensitivity analyses of simple models [22, 39]. There are also large numbers of full and small scale simulation systems used to assess effectiveness of new types of equipment and ways to use them. These microworld and simulation systems are used for off-line analyses to define recommended strategies in conceivably relevant situations.

Systems built for real-time decision-making can take advantage of anytime algorithms with which a coarse prediction can be obtained instantly but is subject to successive refinements when additional time, resources and observations arrive. In such a system, refinements are typically based on either solution improvement or solution re-calculation. An interesting prototype system based on solution improvement is [25] where the situation picture is continuously improved as new observations arrive. The method used is particle filtering, a method where new observations strengthen, weaken or eliminate current hypotheses. A somewhat similar prototype system based on solution re-calculation is [9] where a predicted future situation picture is calculated as a one shot event. Here, solely the particle filtering prediction step is used. The actual choice between the two principles depends on several factors such as the system's intended usage, i.e., whether the decision problem is a one shot problem or a continuous task, and the nature of the problem itself, i.e., whether the present solution can actually be used as basic data for the calculation of a new solution.

Recently, it has become possible to build Bayesian networks to identify the opponent's course of action (COA) from information fusion data using the plan recognition paradigm, which was extended from a single agent context to that of a composite opponent consisting of a hierarchy of partly autonomous units [50]. The conditions for this recognition to work are that the goals and rules of engagement of the opponent are known, and that he has a limited set of COAs to choose from given by the doctrines and rules he adheres to. The opponent's COA can then be deduced reasonably reliably from fused sensor information, such as movements of the participating vehicles. The game component has thus been compiled out of the plan recognition problem. When the goals and resources are not known, these can be modeled as stochastic variables in a BN. However, this is not a strictly correct approach, since the opponent's choice of COA should depend, in an intertwined gaming sense, on what he thinks about our resources, rules of engagement and goals. The situation is essentially a classic Bayesian game, and should be resolved using game algorithms.

3. REALISTIC SITUATION MODELING

It has been suggested that decision-makers often produce simplified and/or misspecified mental representations of interactive decision problems, see, e.g., [34]. Furthermore, most erroneous representations tend to be less complex than the correct ones which, in turn, suggest that decision-makers may act optimally based on simplified and mistaken premises [15]. In this section we discuss and propose the concept of influence diagrams, along with its preliminaries, as a means to specify a reasonably correct representation of the decision problem at hand. An influence diagram is well suited for modeling complex situations. In Section 6, an influence diagram will serve as the underlying model that gives us the basic data needed for the game component.

One goal of artificial intelligence (AI) [45] has been to create expert systems, i.e., systems that can, provided the appropriate domain knowledge, match the performance of human experts. Such systems do not yet exist, other than in highly specific domains, but AI research has inspired important interdisciplinary efforts to solve questions regarding knowledge representation, decision-making, autonomous planning, etc. These results provide a good ground for the construction of C2 decision support systems. Modern expert systems strive for the ideal of a clean separation of its two components; the domain-specific knowledge base and the algorithmic

inference engine [14]. Our work proposes generic inference procedures and, thus, targets the inference engine part of the expert system in this regard. During the last decade, the intelligent agent perspective has led to a view of AI as a system of agents embedded in real environments with continuous sensory inputs. We believe that this is a viable way to reason about C2 decision-making and we adopt the agent perspective throughout this paper.

Agents make decisions based on modeling principles for uncertainty and usefulness in order to achieve the best expected outcome. The assumption that an agent always tries to do its best relative to some utility function, is captured in the concept of rationality. The combination of probability theory, utility theory and rationality constitutes the basis for decision theory. The basic elements that we use for reasoning about uncertainty are random variables. General joint distributions of more than a handful of such variables are impossible to handle efficiently, and modeling distributions as Bayesian networks has become a key tool in many modeling tasks.

A BN offers an alternative representation of a probability distribution with a directed acyclic graph where nodes correspond to the random variables and edges correspond to the causal or statistical relationships between the variables. Calculating the probability of a certain assignment in the full joint probability distribution using a BN means calculating products of probabilities of single variables and conditional probabilities of variables conditioned only on their parents in the graph. The BN representation is often exponentially smaller [45] than the full joint probability distribution table and many inference systems use BNs to represent probabilistic information. Another advantage with the BN representation is that it facilitates the definition of relevant distributions from causal links that are intuitively understandable and, in the case of a dynamic BN, develop with time. Successful(?) uses of these networks include the implementation of the "intelligent paper clip" in Microsoft Office [23], although much of its potential functionality was stripped away in the actual deployment.

An influence diagram is a natural extension to a BN incorporating decision and utility nodes in addition to chance nodes, and represents decision problems for a single agent [24]. Decision nodes represent points where the decision-maker has to choose a particular action. Utility nodes represent terminal nodes where the usefulness for the decision-maker is calculated as a function of the values of its parents. These diagrams can be evaluated bottom up by dynamic programming to obtain a sequence of maximum utility decisions.

When designing decision-theoretic systems to be used for C2 decision-making, complex situations arise where one wants to represent knowledge, causality, and uncertainty at the same time as one wants to reason about the situation, simulating different COAs in order to see the expected usefulness of proposed moves. We

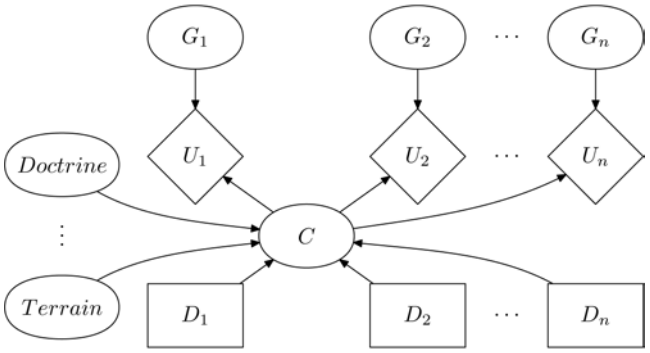


Fig. 1. The C2 process modeled in an influence diagram. Terrain data bases and doctrine are examples of domain-specific subdiagrams that characterize a particular model.

believe the influence diagram is the right choice for both representation and evaluation and propose a simplified schematic generic diagram in Fig. 1 for the C2 process. C is a discrete random variable representing the consequence of the decisions $D_1,\ldots,D_n$. D_1 represents our own decision and $D_2,\ldots,D_n$ represent the decisions of the other agents. G_1 is a discrete random variable that represents our own goals. U_1 is the utility that we gain after performing decision D_1 depending on the consequence C and our own goals G_1. G_i and U_i are defined similarly for the other agents where $2 \leq i \leq n$.

The diagram in Fig. 1 is a simplified representation, to be connected to models—encoded as BNs—of terrain, doctrine, etc., that can be implemented as subdiagrams with causal relationships between different nodes of models. While these subdiagrams are interesting in their own right, they are not the topic of this article. Hence, we have chosen to think of them as existing models that influence the decisions we are modeling.

A problem with the diagram in Fig. 1 is that it does not capture "gaming situations" where one wants to reason about opposing agents that act according to beliefs about one's own actions. Such dependencies are not possible to model in an influence diagram or BN without additional machinery. At this point it should also be noted that the diagram in Fig. 1 should not be considered to be very useful in its own right. Rather, it is a statement of the problem we are trying to solve. Among other things, the diagram is not regular which is a requirement for algorithms that evaluate influence diagrams, see, e.g., [46]. Regularity assumes a total ordering of all of the decisions, a reasonable condition for a single decision-maker who only needs to take his own actions into account.

In this work we use the influence diagram as basic data to develop a generalized technique that solves problems for multiple decision-makers. In Fig. 2 we give an alternative algorithm for evaluation of influence diagrams with multiple agents, inspired from the single agent construction found in [44, 45]. Here, the payoffs for all combinations of alternatives are returned instead of only the alternative with the highest possible payoff.

Input: Influence diagram with decision nodes $D_1, \ldots, D_n$ and utility nodes $U_1, \ldots, U_n$ belonging to agents $1, \ldots, n$ respectively.

Output: An n-dimensional utility matrix A containing, for each possible value combination on the respective agents' decision nodes, the resulting utility vector $(u_1, \ldots, u_n)$.

1. Set the evidence variables for the current world state, i.e., use the percepts that the agent has received to date to assign values to a subset of the random variables in the influence diagram.
2. For each possible value combination of the decision nodes;
   (a) Set the decision nodes to their respective values.
   (b) Calculate the posterior probabilities to the parent nodes of the utility nodes using a standard probabilistic inference algorithm or by the node elimination method of [46].
   (c) Calculate the resulting utility vector for the action combination and store this in the utility matrix A.
3. Return the utility matrix A.

Fig. 2. Algorithm for evaluating an influence diagram where multiple agents make decisions.

4. AGENT INTERACTION

The decision situation that arises in decision node D_1 in the influence diagram depicted in Fig. 1 is characterized by its dependency on other actors' decisions. Standard AI tools for solving decision-making problems in complex situations, such as dynamic decision networks and influence diagrams, are not applicable for these kinds of situations, as the decisions are intimately related to the other agents' decisions. Game theory, on the other hand, provides a mathematical framework designed for the analysis of agent interaction under the assumption of rationality where one tries to identify the game equilibria as opposed to traditional utility maximization principles. A game component in multi-agent decision-making thus uses rationality as a tool to predict the behavior of other agents.

In higher level C2, i.e., threat prediction in a data fusion context, the need of a game component becomes obvious [55]. Circular relationships are not allowed in influence diagrams or other traditional agent modeling techniques and therefore we cannot make the agents' decisions dependent on each other in the diagram in Fig. 1. On lower level C2 this need is not as obvious, because agents' choices are to a large extent driven by standard operating procedures obtained by training and developed using off-line game analyses. On this level, like in helicopter dogfights, successful developments of strategies have been obtained with look-ahead in extensive form, i.e., perfect information game trees with zero-sum payoffs as reported in [27] or moving horizon imperfect information game trees as reported in [54]. The depth of the game tree corresponds to inference of agents' actions that are dependent on each other, i.e., a series of what-if questions such as "what is the usefulness if agent i performs action c_i and the other agents perform actions $c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n$ which in turn makes agent i respond with action c_i'," etc. Look-ahead algorithms are typically modeled using a discount factor $\gamma \in (0,1)$ that reduces the utility by γ^d where d is the tree depth. For problems in which the discount factor is not too close to 1, a shallow search is often good enough to give near-optimal decisions [45].

Look-ahead game trees have been used successfully for reasoning in, possibly uncertain, games with perfect information where optimal solutions are obtained with the minimax algorithm. Examples of such games are chess, go, backgammon, and monopoly. In the context of C2 we deal with imperfect information which forces us to solve a more complex game, more similar to poker, since we cannot be sure of exactly where we are in the game tree. Although ordinary minimax algorithms cannot be used in our context it is still likely that the ideas from ordinary game play algorithms, such as the famous alpha-beta pruning [29], can be re-used to some extent. This is interesting as these ideas rest on almost a century of research and experience [33, 45].

Decision-making in environments where multiple agents make decisions based on what they think the other agents might do is a difficult problem, and the use of game theory for agent design has so far been limited due to lack of standard implementation methods. We believe, however, that this barrier will be overcome as more research is focused on the use of game theory for agent design. The widely used AI book by Russell and Norvig [45] added a section on game theory just recently which indicates that the ideas are new and still need to be investigated more thoroughly.

One of the barriers that do exist when using traditional game theory for agent design is that it assumes that a player will definitely play a (Nash) equilibrium strategy. This assumption is certainly true in applications where the game is a designed mechanism, such as the management of (own) mobile sensors [26, 57] or the construction of algorithms for efficient network capacity sharing [4]. However, these situations must be considered a small subset compared with the many situations in everyday life that involve uncertainty about both the other actors and the world as a whole. Over time it has come to be recognized that benevolence is the exception; self-interest is the norm [43]. Particularly, in our C2 application self-interest is the norm that commander training seeks to foster. In this work we aim at solving this problem using the Bayesian game technique, which is described below.

Other problems with game theory for agent design are the lack of methods for combining game theory with traditional agent control strategies [45] and the lack of standard computational techniques for game-theoretic reasoning [33].

In this paper we propose the use of a Bayesian game for modeling higher-level agent interaction in an attempt to obtain better situation awareness in a C2 system. As situation awareness is obtained using fusion techniques we believe that the game component is an integral part of the data fusion process and provides information that

is needed in level three data fusion processing according to the JDL model [37, 47, 56]. A Bayesian game is a game with incomplete information, that is, at the start of the game the players may have private information about the game that the others do not know of. Also, each player expresses its prior belief about the other players as a probability distribution over what private information the other players might possess.

5. STRUCTURE OF GAMES AND THEIR REPRESENTATIONS

Recent developments in game theory and AI have made applications with significant game components feasible. Most of the work, however, does not address Bayesian games. Many description methods have been developed with algorithmic techniques being able to solve quite large games if they are of the right type. The extensive form of a game is a tree structure, where a non-terminal node can describe a chance move by nature (random draw) or a move possible for one of the participants, and a leaf node represents the end of the game and its payoff after evolving through the path to it. The immediate descendants of a non-leaf represent the alternative outcomes of a chance move (in which case the node is associated with a probability distribution) or the set of actions available for the player in turn at this point. This is adequate for leisure games like chess, a perfect information game, but the chess game tree does not fit into any computer. A deterministic game with full information (like generalized chess or checkers) can be solved if its game tree can be traversed, by bottom-up dynamic programming.

In games with imperfect information, the exact position in the game tree may not be known to players. This is the case in leisure games of cards, where the hand of a player is only available to her. The determination of optimal strategies must use a game tree where the decision is the same for a whole information set, a set of nodes for a player where the information available to her is the same. As an example, at the first bid of a game of contract bridge, each of the possible distributions of the cards not seen by the player is in the same information set. Bottom-up evaluation does not work, because at the lower levels of the game tree the players have information on the hidden information that was communicated by their opponents' choices of moves (like the initial round of bidding in bridge). This situation is solved by putting the game on strategic form, which means that all combinations of moves for all of a player's information-equivalent nodes in the tree, and all chance moves, are listed with their payoffs. Solutions can be found with numerical methods, linear programming techniques for zero-sum games [11] and solution methods for the linear complementarity problem (LCP) for general games [13]. For the former, a unique mixed (randomized) strategy for each player is a non-controversial definition of the game's solution. For the latter, the Nash equilibrium is the accepted solution concept [42]. A Nash equilibrium always, under general assumptions, exists but is less non-controversial since sometimes several equilibria exist, and there are alternative proposals regarding how to find one that is in a tangible way more relevant than the others. The payoff matrix is typically impossibly large, and games of this type, like standard variants of poker and bridge, have no known optimal solution although interesting approximation algorithms have appeared recently [5]. In the above games, all players know the exact structure and payoff system of the game. This is adequate for many purposes, but not for our application.

The concept of a Bayesian game is fairly complex and different views abound in the literature. With notation from [41], a Bayesian game, Γ^{b}, is defined by

$$\Gamma^{\mathrm{b}} = (N, (C_i)_{i\in N}, (T_i)_{i\in N}, (p_i)_{i\in N}, (u_i)_{i\in N}) \qquad (1)$$

where N is a set of players, C_i is the set of possible actions for player $i \in N$, T_i is the set of player i's possible types, p_i is a probability distribution representing what player i believes about the other players' types, and u_i is a utility function mapping each possible combination of actions and types into the payoff for player i. It should be noted that the set notation we use differs from standard mathematical notation. Indices contain one or several players in the set N and hence represent the "player dimension." When there is no subscript at all we actually mean a set with a variable for each player in N which is denoted a profile. The subscript $-i$ denotes the set of all players except for player i, i.e., $N \setminus \{i\}$. The other dimension is defined by the letter itself that can be either lower-case, representing one particular choice, or upper-case, representing the set of all possible choices. Henceforth, C_i is the set of possible actions for player i, $c_i \in C_i$ is one of player i's possible actions, $c \in C$ is a possible strategy profile in the game, and C is the set of all possible strategy profiles that we may encounter in the game.

The definition given above is a flat representation given originally in [21]. It seems as if it only states first-order beliefs of players about each other, but this is not a fair perspective. We want to consider all types of higher-order knowledge, such as what player 1 believes that player 2 believes that player 1 ... believes. This type of information can indeed be modeled in a standard Bayesian game, under quite general conditions, as shown in a strictly mathematical and non-algorithmic argument in [38]. On the other hand, the amount of information required to perform such modeling can be infinite and thus not extractable from, or actually used by, experts and decision-makers. Bayesian games can have infinite type sets even in simple cases like natural analyses of bargaining situations. We will restrict our attention to games with finite type sets and players, since otherwise general solution algorithms do not exist (games with infinite type sets must be analyzed manually to bring about a finite solution algorithm).

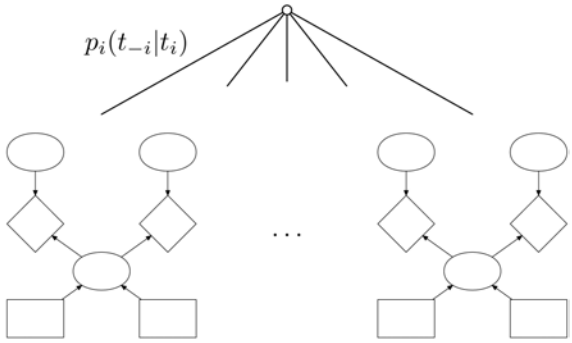


Fig. 3. Architecture overview. Models are represented by influence diagrams that yield payoff values for a Bayesian game.

An important class of Bayesian games is games with consistent beliefs. In this case the player's belief, conditional on his type, about other players' types are all derivable from a global distribution over all players' types by conditioning, i.e., $p_i(t_{-i} \mid t_i) = p(t_{-i} \mid t_i)$. Hence, this class is a subclass of imperfect information games. The assumption of consistent beliefs is both required and natural for most applications; it simply means we should model the players using all information we currently have in our possession. Although game theory means we solve the game for all players at the same time, the solution is still obtained from one particular decision-maker's view of the situation. Therefore, consistent versus inconsistent beliefs becomes more of a philosophical question and we will assume consistent beliefs throughout this work.

Inputs: 1) A list of influence diagrams; one influence diagram for each possible agent model that needs to be considered. Decision nodes $D_1, \ldots, D_n$ and utility nodes $U_1, \ldots, U_n$, belonging to agents $1, \ldots, n$ respectively, need to exist in all diagrams. 2) A common prior probability distribution P over the possible agent models.

Output: Solution proposals for the influence diagram decision variables $D_1, \ldots, D_n$ in the form of mixed strategy Nash equilibria.

1. Let each influence diagram correspond to a Bayesian game type profile $t \in T$, representing that each influence diagram corresponds to different beliefs regarding the participating agents' private information.
2. Formulate the Bayesian game

$$\Gamma^{\mathrm{b}} = \left(N, (C_i)_{i \in N}, (T_i)_{i \in N}, (p_i)_{i \in N}, (u_i)_{i \in N}\right)$$

so that;

   (a) N, the set of players, corresponds directly to the set of participating agents,

   (b) C_i corresponds to the set of actions available to agent i in decision node D_i in the influence diagrams,

   (c) T_i contains the possible types for player i; induced by T according to item 1 above,

   (d)

$$p_i(t_{-i} \mid t_i) = \frac{P(t)}{\sum\limits_{s_{-i} \in T_{-i}} P(s_{-i}, t_i)} \quad \text{(consistent beliefs)},$$

   (e) $u_i \colon C \times T \to \mathbf{R}$ is given by the algorithm in Fig. 2, i.e., the algorithm in Fig. 2 needs to be run for each type profile $t \in T$ to obtain the respective utilities given a certain model.

3. Calculate one or more solutions to the Bayesian game in the form of mixed strategy Nash equilibria.
4. Equilibria in the game correspond directly to solution probability distributions over the decision variables $D_1, \ldots, D_n$ in the original influence diagrams. These distributions are returned as solution concepts—not necessarily to be executed, but to further enhance a commander's predictive situation awareness.

Fig. 4. Summary of the game component.

6. THE GAME COMPONENT

In this section we define the proposed information fusion game component using notation from [41]. A brief concept sketch is given in Fig. 3 and a more formal summary is given in Fig. 4 which, in turn, uses the algorithm depicted in Fig. 2. The objective has been to specify an architecture that is suitable for threat prediction in the C2 domain. The most important criteria for the specification of such an architecture are that the agents' decisions are based on their belief regarding the other agents' private information, and that the architecture is made up from an underlying well-established and realistic probabilistic model of the situation. We achieve the former criterion by the use of a game with incomplete information, and the latter criterion by using an influence diagram for representing our model of the current situation awareness.

A top-down perspective on the architecture can be seen in Fig. 3, depicting a probability distribution over the possible worlds. Each such world is modeled in an influence diagram, such as the diagram outlined in Fig. 1, containing nodes for the goals (G_i), the possible courses of actions (D_i), and the payoff (U_i) for each respective agent. Apart from these variables, each influence diagram is connected to model specific subdiagrams containing environmental descriptions, doctrine and other properties specific to the model in question. An important observation regarding the model in Fig. 1 that motivates the use of game theory is the fact that this model, seen as an ordinary influence diagram, does not account for situations when agents' try to make decisions that are influenced by other agents' decisions. That is, it is not capable of representing circular causal relationships between D_1 and D_2. To account for this gaming perspective we therefore think of the possible world states as Bayesian game type profiles. Utilities are obtained for each such type profile by using its correlated influence diagram to create a strategic form game, i.e., utilization of the algorithm in Fig. 2 which for each combination of the decision profile $D_1, \ldots, D_n$ calculates utilities $U_1, \ldots, U_n$.

Using our prior belief regarding which model is accurate, we then obtain a Bayesian game for the whole decision problem. Calculation of equilibria in the Bayesian game yields solutions for the decision variables $D_1, \ldots, D_n$ in the form of mixed strategy Nash equilibria. A more formal description of the scheme can be found in Fig. 4.

Assuming consistent beliefs, the solution to a Bayesian game is obtained by introducing a new root node called a historical chance node that is used to imple-

ment the Bayesian property of the game. A historical chance node differs from an ordinary chance node in that the outcome of this node has already occurred and is partially known to the players when the game model is formulated and analyzed. For each set of possible types, the edges from the root node in the game correspond to the model that is used if the players were of this type. We say that a player i believes that the other players' type profile is $t_{-i} \in T_{-i}$ with subjective probability $p_i(t_{-i} \mid t_i)$ given that player i is of type $t_i \in T_i$. Again, note that the subscript $-i$ is standard notation for the set of all players except for player i, i.e., t_{-i} is a list of types for all the other players.

For each type profile $t \in T$, an influence diagram, as in Fig. 1, describes the decision situation using random state variables. The different models differ in properties that cannot be seen in Fig. 1, consisting of other random variables describing for example terrain, doctrine, and belief regarding all kinds of properties that do not rely on other participating agents' decisions. In the context of our Bayesian C2 game the historical chance node is thus a lottery over the possible models that are represented as influence diagrams.

The Bayesian property of the game might seem trivial at first glance, but the historical chance node at the root of the tree poses a serious concern to us. To establish Nash equilibria for the game the normal representation in strategic form is needed, but the algorithm for the creation of this relies on the players being able to decide their strategies before the game begins, which is not true in a Bayesian game that is represented with a historical chance node. The solution, due to Harsanyi [21], is to reduce the game to Bayesian form and compute its Bayesian equilibria. Such an equilibrium consists of a probability distribution over actions for each player and each of this player's types. This can in principle be accomplished by solving an LCP to obtain a mixed strategy for each type of each player. Although in game-theoretic studies, Bayesian games are often defined with infinite type and action spaces, we classify actions discretely after doctrines the players are trained to follow, and if the intuitive type of a player is a continuous variable we discretize it.

At level two, for each node represented by a distinct type profile $t_{-i} \in T_{-i}$, the node is the start of the model that the type profile $t_{-i} \in T_{-i}$ gives rise to. To represent this model we use a game on strategic form; that is, a game with players N, actions $(C_i)_{i \in N}$, and utility functions $(u_i)_{i \in N}$.

The (still Bayesian) game relates to the influence diagram in Fig. 1 in that N represents the n agents that are about to make decisions $D_1, \ldots, D_n$, C_i represents the actions available for agent i in decision node D_i, and u_i is the utility that is obtained in the diamond shaped utility node U_i which is, in turn, depending on the random variables C and G_i denoting the world consequence and the agent's goals respectively.

7. EQUILIBRIA AND COMPLEXITY

While modeling and representing a C2 situation is interesting in its own right, a primary concern is the use and interpretation of the model. In game theory the concept of Nash equilibria defines game solutions in the form of strategy profiles in which no agent has an incentive to deviate from the specified strategy. Without doubt, defining equilibria is the foremost goal in game theory. Fortunately, this means that we can lean on well-established results in our effort to find equilibria for the C2 situation.

For a Bayesian game, Harsanyi [21] defined the Bayesian equilibrium to be any set of mixed strategies for each type of each player, such that each type of each player would be maximizing his own expected utility given that he knows his own type but does not know the other players' types. Mathematically speaking, a Bayesian equilibrium for a Bayesian game Γ^{b}, as defined in (1), is any mixed strategy profile σ such that, for every player $i \in N$ and every type $t_i \in T_i$,

$$\sigma_i(\cdot \mid t_i) \in \arg\max_{\tau_i \in \Delta(C_i)} \sum_{t_{-i} \in T_{-i}} p_i(t_{-i} \mid t_i) \times \sum_{c \in C} \left(\prod_{j \in N-i} \sigma_j(c_j \mid t_j) \right) \tau_i(c_i) u_i(c,t). \quad (2)$$

Here, $\Delta(C_i)$ denotes the set of probability distributions over the set C_i, i.e., the set of possible mixed strategies that player i can choose from, and $\sigma_i(\cdot \mid t_i)$ is the, possibly mixed, strategy of player i in type t_i.

Existence of a Bayesian equilibrium solution in mixed strategies follows from the famous existence theorem for general games, which is due to Nash [42]. Solution methods for general-sum game-theoretic problems are however intractable for the generic case. The most well-known solution method, the Lemke-Howson algorithm [36, 49], solves a linear complementarity problem [13]. The computational complexity for finding one equilibrium is still unclear. We know, according to Nash's theorem [42], that at least one equilibrium in mixed strategies exists but it is problematic to construct one. The Lemke-Howson algorithm exhibits exponential worst case running time for some, even zero-sum, games. However, this does not seem to be the typical case [49]. Interior point methods that are provably polynomial are not known for linear complementarity problems arising from games [49]. Methods amounting to examining all equilibria, such as finding an equilibrium with maximum payoff, have unfortunately been proven **NP**-hard [19], so for these kinds of problems no efficient algorithm is likely to exist.

The standard way of calculating equilibria in a game in extensive form is to transform the game into strategic form. However, the creation of the matrix for the strategic form typically causes a combinatorial explosion. This is due to each value in the matrix represen-

tation of a strategic form game representing the payoff for a complete strategy. Hence, even though a game tree typically contains widely different decision alternatives in different subtrees the decisions in the other subtree still need to be considered. Therefore the strategic form matrix dimension grows for each node that is traversed. In a series of articles [30, 31, 48] published during the last decade the sequential form as a replacement for the strategic form has provided a representation suitable for efficient computation of equilibria in an extensive imperfect game with chance nodes. The idea is to replace the game's strategies with new strategies based on sequences ranging from the root node down to the leaves. That is, each sequence represents a possible course of events in the game. As the creation of the matrix for the sequence form relies on payoffs that are already in the tree the problem complexity is reduced from a **PSPACE**-complete problem into a problem that is linear in the size of the tree. However, it should be kept in mind that general game trees often share decision alternatives and, hence, do not exhibit a full scale combinatorial explosion. In totally symmetric problems, as investigated in for example [8], the choice of game representation therefore does not affect the computational tractability significantly. Also, as mentioned above a pre-requisite for the sequential method to be effective is that the game is in extensive form to start with. Referring to the information fusion game component, as outlined in Section 6, this is problematic since the algorithm depicted in Fig. 2 results in a strategic game. However, using an additional chance node denoting the common model prior, it is possible to hinder this combinatorial explosion by transforming the whole game component into one large influence diagram. This influence diagram can then be utilized to create the game tree directly using the multi-agent influence diagram conversion algorithm in [32] which, in turn, is a straightforward extension of the single-agent decision tree algorithm found in [44].

As indicated, the incentive for us to actually use the sequential method when developing the information fusion game component has so far been limited, but the relation between the sequential method and its potential savings must be kept in mind when developing the game component further. A model incorporating a series of ordered decisions, or perhaps a hierarchy of decisions as outlined in [7], is likely to benefit significantly from this representation. More information on this topic regarding so-called MAIDs, an acronym for multi-agent influence diagrams, and their relation to the information fusion game component can be found in Section 10.

Although game-theoretic methods are, in most cases, computationally infeasible in theory, computation of optimal solutions still seems to be tractable in reasonably sized C2 decision problems [8]. Moreover, despite the intractability of finding all optimal solutions there exist fast algorithms that often finds all, or nearly all, solutions.

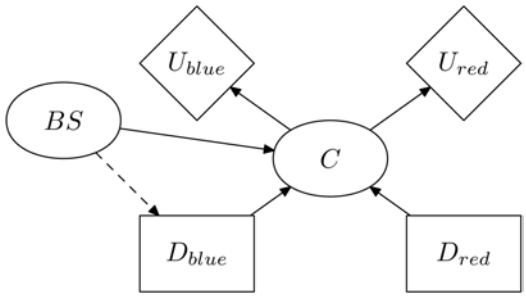


Fig. 5. Influence diagram depicting an example scenario with a blue player and a red player. The Boolean node *BS* denotes the blue player's private information that gives rise to two blue player types in the game.

8. A SMALL EXAMPLE

In this section the gaming perspective is illustrated with an example of a situation where the commander wishes to reason about two possible models.

At a certain point in battle, a blue (male) unit controls an asset (equipment or territory). When a red (female) unit appears on the scene the blue unit knows immediately whether its own forces are inferior or superior. The red unit on the other hand, does not know anything regarding the capabilities of the blue unit. The blue unit has the choice to engage in battle or to remain passive. If he remains passive the red unit will use her sensors to detect whether he is superior or not and if he is inferior she will force him to give up the asset. On the other hand, if the blue unit chooses to engage the red unit she will be faced with an opportunity to retreat or to engage. If the blue unit is superior and the red unit chooses to engage him, he will both defeat the red unit and keep control of the asset. If the blue unit is inferior and the red unit chooses to engage him he will lose both the battle and the asset. If the red unit retreats the blue unit will keep control of the asset whether he is superior or not. The central part of the corresponding influence diagram is shown in Fig. 5. The random variable *BS* (Blue Superior) constitutes evidence for the blue decision-maker but not for the red decision-maker, denoted with the dotted arrow from *BS* to D_{blue}. The node *BS* is also a parent to the world consequence node *C* because it determines the outcome of an engagement and thus the state of the world. The *C* node then affects the decision-makers' respective utility nodes where, in this case, $U_{\text{blue}} = -U_{\text{red}}$ since the game is zero-sum. It is vital to understand the difference between evidence variables and query variables to fully grasp the example (and the game component as a whole). For the blue player, the variable *BS* is evidence which, in turn, gives rise to one "blue superior game model" and one "blue inferior game model." For the red player, *BS* is just an ordinary random variable with an associated conditional probability table. The chance node *C*, on the other hand, can never have its value set as an evidence variable as it is referring to a future state.

TABLE I
Payoff Matrices for the Myerson Card Game

t_1 = 1.a (superior)

Player 1 \ Player 2	M	P
R	$2,-2$	$1,-1$
F	$1,-1$	$1,-1$

t_1 = 1.b (inferior)

Player 1 \ Player 2	M	P
R	$-2,2$	$1,-1$
F	$-1,1$	$-1,\ 1$

If the value of 1) winning the battle and 2) controlling the asset are worth one utility unit respectively, the game becomes similar to the card game of Myerson [41]. As indicated in the situation description, we follow the convention that odd-numbered players are male and even-numbered players are female. This is common practice in game theory and has no deeper meaning. At the beginning of the game both players put a dollar (the asset) in the pot. Player 1 (the blue force) looks at a card from a shuffled deck which may be red (he is superior) or black (he is inferior). Player 2 (the red force), on the other hand, does not know the color of the card but maintains a belief of this in the form of a probability distribution in her influence diagram, i.e., a belief of the possibility of player 1 being superior or inferior. Player 1 moves first and has the opportunity to fold (F) or to raise (R) with another dollar, i.e., remain passive or engage in battle. If he raises, player 2 has the opportunity to pass (P) or to meet (M) with another dollar in the pot, i.e., retreat or engage in battle.

We let $\alpha \in (0,1)$ denote player 2's belief of player 1 being superior. In this example, player 1 also knows the value of α, i.e., the players' beliefs are consistent. The situation can then be modeled with a Bayesian game Γ^{b}, as defined in (1), with $N = \{1,2\}$, $C_1 = \{F,R\}$, $C_2 = \{M,P\}$, $T_1 = \{1.\mathrm{a}, 1.\mathrm{b}\}$, $T_2 = \{2\}$, $p_1(2 \mid 1.\mathrm{a}) = p_1(2 \mid 1.\mathrm{b}) = 1$, $p_2(1.\mathrm{a} \mid 2) = \alpha$, $p_2(1.\mathrm{b} \mid 2) = 1-\alpha$ and $(u_1(c_1,c_2,t_1), u_2(c_1,c_2,t_1))$ as in Table I.

Solving the game using the technique described by Harsanyi [21] involves introducing a historical chance node, a "move of nature," that determines player 1's type, hence transforming player 2's incomplete information regarding player 1 into imperfect information. The Bayesian equilibrium of the game is then precisely the Nash equilibrium of this imperfect information game. The Harsanyi transformation of Γ^{b} is depicted in Fig. 6 on extensive form.

Note that there are two decision nodes denoted "2.0" that belong to the same information set, representing the uncertainty of player 2 regarding player 1's type. Also, note that the move labels on the branch following the "1.a" node do not match the move labels on the branches following the "1.b" node, representing that player 1 is able to distinguish between these two nodes. The normal way of solving such a game is to look at the strategic representation, as seen in Table II.

In order to solve the game, first note that Fr is dominated by Rr and that Ff is dominated by Rf

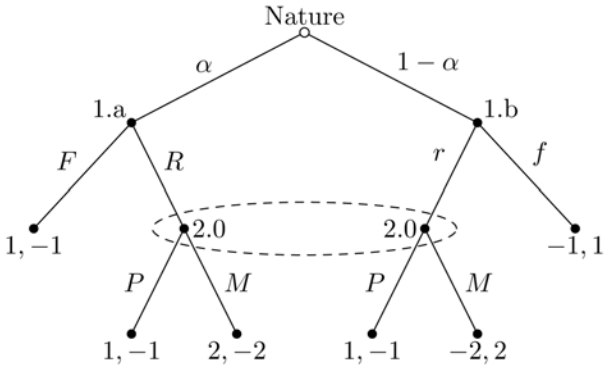


Fig. 6. The Harsanyi transformation of the game in Table I.

TABLE II
The Strategic Form of the Game in Fig. 6

Player 1 \ Player 2	M	P
Rr	$4\alpha-2, 2-4\alpha$	$1,-1$
Rf	$3\alpha-1, 1-3\alpha$	$2\alpha-1, 1-2\alpha$
Fr	$3\alpha-2, 2-3\alpha$	$1,-1$
Ff	$2\alpha-1, 1-2\alpha$	$2\alpha-1, 1-2\alpha$

regardless of the value of α, i.e., player 1 will always raise if in a superior position. Second, if $3/4 \le \alpha < 1$ we have that P dominates M so that player 2 will always choose to pass, which, in turn, implies that player 1 will always choose to raise. Hence, $([Rr],[P])$ is the one and only equilibrium strategy profile for $3/4 \le \alpha < 1$. For $0 < \alpha < 3/4$ there are no equilibria in pure strategies (just check all four remaining possibilities) and we have to look for equilibria in mixed strategies. Let $q[Rr] + (1-q)[Rf]$ and $s[M] + (1-s)[P]$ denote the equilibrium strategies for players 1 and 2 respectively, where q denotes the probability that player 1 raises with a losing card and s the probability that player 2 meets if player 1 raises. A requirement for an equilibrium for player 1 is that his expected payoff is the same for both Rr and Rf, i.e., $s(4\alpha-2) + (1-s)1 = s(3\alpha-1) + (1-s)(2\alpha-1) \Rightarrow s = 2/3$. Similarly, to make player 2 willing to randomize between M and P, M and P must give her the same expected utility against $q[Rr] + (1-q)[Rf]$ so that $q(4\alpha-2) + (1-q)(3\alpha-1) = q1 + (1-q)(2\alpha-1) \Rightarrow q = -\alpha/(3(\alpha-1))$.

We can now use the equilibrium strategy of the imperfect information game in order to derive the Bayesian equilibrium of the game Γ^{b}. A Bayesian equilibrium specifies a randomized strategy profile containing one strategy $\sigma_i(\cdot \mid t_i)$ for all combinations of players and types. Hence, the unique Bayesian equilibrium of the game Γ^{b} is $\sigma_1(\cdot \mid 1.\mathrm{a}) = [R]$, $\sigma_1(\cdot \mid 1.\mathrm{b}) = q[R] + (1-q)[F]$, $\sigma_2(\cdot \mid 2) = 2/3[M] + 1/3[P]$ for $0 < \alpha < 3/4$ and $\sigma_1(\cdot \mid 1.\mathrm{a}) = [R]$, $\sigma_1(\cdot \mid 1.\mathrm{b}) = [R]$, $\sigma_2(\cdot \mid 2) = [P]$ for $3/4 \le \alpha < 1$.

Although this simple game presents a solution that is not entirely trivial, it is simpler than our full family of games in that it is zero-sum with only two players and thus has a unique Nash equilibrium that is computationally easy to find.

9. SOLUTION INTERPRETATION

Nash equilibria, in the form of mixed strategies, as a solution to decision problems require a moment of thought. On the one hand, it is easy to argue that the equilibrium strategy is theoretically sensible. After all, the notion of Nash equilibria, building on the concept of rationality, defines precisely this. By using the idea of Bayesian games we are able to create alternative models regarding agents that are in some way "irrational." Thus, by using Bayesian games we can counterattack any objections on the existing model by simply extending the model with a new submodel that models the objection in question. Of course, this also requires assigning a prior probability to the new submodel and re-evaluating the prior probabilities for the existing submodels, which makes sense if someone comes up with an objection (which is interpreted as a new model that we have not thought of before). If the objection is independent of the existing models, normalization is the natural way to re-assign probabilities. Otherwise it is natural to let the prior probability of the new model be represented by a reduction of prior probabilities of the model or the models that it depends on. In most cases we believe that it is appropriate to have a separate model for the "uncertain case" that takes care of whatever we have not thought of. In that case the new submodel, provided it is independent of other existing models, typically reduces our overall uncertainty regarding the situation and thus causes a reduction of prior probability for the earlier mentioned "uncertain case" submodel. Models that take care of the rest, i.e., that represent options or possibilities that we are not yet aware of, are often found in proposed architectures for multi-agent modeling, see for example [20] where irrational behavior as well as lack of information is modeled in so called "no information models."

On the other hand, although representing the theoretically rational course of action, the Nash equilibrium poses several concerns regarding its interpretation. Looking at the example scenario in Section 8, it is interesting to see how q and s varies depending on α which is shown in the diagram in Fig. 7, i.e., how the solution to our decision problem varies depending on our subjective beliefs regarding the opponent being superior or inferior. How do we convince a commander that he should decide what to do by throwing a die that varies depending on $q(\alpha)$? He probably understands that he is bluffing, and that it is in general disadvantageous both to always bluff and to never bluff. Without knowing the background to the solution it is not trivial to understand why player 1 should raise with a losing card with probability $q(\alpha)$ in Fig. 7. Perhaps even more strange is that player 2's counterattack, the probability $s(\alpha)$ to meet, is kept constant at $s(\alpha) = 2/3$ until $\alpha = 3/4$ when it suddenly goes down to zero. So there is a discontinuity in the optimal strategy when α varies, although at the discontinuity the optimal utilities vary continuously. Hence, an error in the α estimate has no large utility effect although the equilibrium solution strategies may vary significantly. The conclusion regarding the Myerson card game is that a simple problem gives us a solution that is difficult to understand intuitively and that may or may not, dependent upon the decision-maker's objective, raise questions regarding robustness. This is quite typical, see for example [6] for another example, and we need to address the question of how to use the solution in a sensible way. To actually throw the die is part of the solution and if this is not performed the commander is not rational and, hence, will be outperformed by a rational opponent that is capable of modeling this behavior. It is probably easier to accept the opponent's randomized strategy as a prediction. Then the optimality of one's own randomized strategy is fairly easy to establish. As can be seen in Fig. 7, however, such a prediction must be analyzed for discontinuities that indicate potential issues related to strategy robustness.

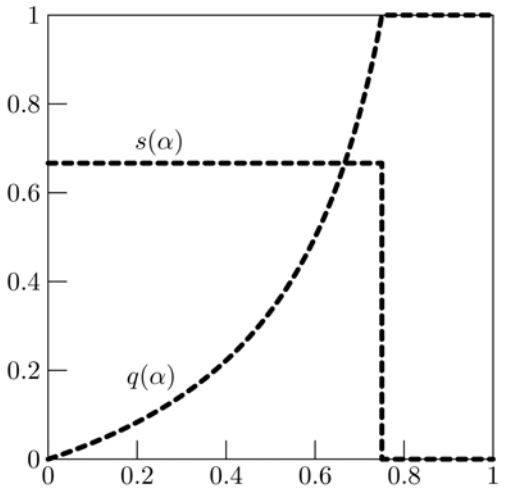


Fig. 7. The graph shows how the game-theoretic solution $s(\alpha)$ to the decision problem in Section 8 varies in a non-intuitive manner depending on the player's speculation regarding the other player being inferior or superior.

To outperform someone by exploiting his plan is called outguessing. It is tempting to use an estimate of the risk of exploitation as a basis for decision-making so that the (risk-compensating) Nash equilibrium mixed strategy is chosen when the risk is high and the pure strategy with the highest payoff is chosen when the risk is low. An approach in this direction using hypergame theory, which is fundamentally heretical to the concepts of game theory, is proposed in [53].

10. RELATED WORK

Development of game tools is an active area in AI. In the Gala system of [33], tools exist for defining games with imperfect information. A tractable way to handle games with recursive interaction in strategic form was developed in [20], where the potentially infinite recursion of beliefs about opponents is represented approximately as a finite depth discrete utility/probability matrix tree defining the players' beliefs about each other. The solution emerging from this modeling is not a Bayesian game equilibrium, however.

There is a significant body of work on multi-agent interactions in the intelligent agents literature. A survey of methodological and philosophical problems appears in [20]. The principle of bounded rationality can be taken as an excuse to use simpler solution concepts than Bayesian game solutions. In our case, there is no reason to assume that the opponent is not rational—there would be few excuses if he turned out to be so. This does not mean that it is not necessary to take advantage of opponents' mistakes when they occur. Plans must foresee this and have opportunities of opponent mistakes as a part; but these options should not be executed until the evidence of the mistake is sufficient. The recursive modeling of multi-agent interaction of [20] (mostly developed for cooperative rather than competitive interaction) is thus not appropriate in our application. The proposal in [27] is to use game theory with zero-sum game tree look-ahead for C2 applications. Although this approach was successful for analysis of lower level game situations, we have argued above that it is not enough in a complete higher level C2 tool.

In [32] the concept of a multi-agent influence diagram (MAID) is defined, which in a similar manner to our information fusion game component partitions the decision and utility variables by agent so that utilities and decisions of many agents can be described. The key idea behind the MAID framework is to use the graph structure to explicitly state strategic relevance between decision variables which, in turn, is being used to break up a large game into a set of singly connected components (SCCs) which can be solved in sequence. The complexity of equilibria computation in the full game is therefore reduced to the complexity of equilibria computation in the largest SCC in the MAID. In some games, where the maximal size of an SCC is much smaller than the total number of decision variables, the MAID representation provides exponential savings over existing solution algorithms. In the worst case, however, the strategic relevance graph forms a single large SCC and the MAID algorithm simply solves the game in its entirety, with no computational benefits. The influence diagrams in the information fusion game component outlined in Section 6 are unfortunately examples of such large SCCs. As it turns out, the whole game component could be alternatively represented by a MAID with a single large SCC provided an additional chance node, representing the "move of nature," was added to connect the models to each other.

An extension of the MAID framework is the NID—Network of Influence Diagrams. In the version described in [18], several MAIDs—or other game representations—can be connected in a directed acyclic graph, where outgoing arcs are labeled with a probability distribution. This allows us to define situations where agents do not all use the same model, but there is no way to describe in an acyclic graph a situation where there is mutual uncertainty and inconsistent beliefs about the game structure and the opponents' goals.

11. CONCLUSIONS

In higher level command and control (C2) we can be certain that large efforts are directed towards predicting the beliefs, desires, and intentions of the adversary—and there will not be a common agreed upon model of the situation and its utilities. In fact, the complex nature of any C2 decision situation makes it necessary to go beyond any proposed theoretical model and question how, if at all, it can be used in practice. Adding conflict, where opposing parties try to outguess each other, complicates things even further with the necessary addition of a gaming perspective—putting stress on a decision situation that is complex already from the beginning.

In this paper we propose a way to overcome the barriers between theory and practice, taking into account opponent modeling as well as current state-of-the-art C2 situation modeling principles. We characterize the proposed architecture as an information fusion game component to emphasize the inherent dependencies between the gaming perspective and the process of fusing sensor data into a comprehensible situation picture. It is our belief that game theory should not be considered just another tool in the decision-maker's toolbox. Rather, it is the science of agent interaction itself, i.e., we consider game theory to be the whole toolbox as well as a statement of the information fusion threat prediction problem.

Game-theoretic tools have a potential for situation prediction that takes uncertainties in enemy plans and deception possibilities into consideration. The idea behind Bayesian games is particularly interesting, and needed, from the viewpoint of a commander facing a real setting decision problem; it combines several models of the situation, thus making it possible to consider such diverse factors as opponent irrationality or the decision-maker's intuition by incorporating these ideas as separate models. However, Bayesian games, as well as game theory in general, still have shortcomings when representing realistic, potentially large and complex, situation descriptions—at least compared to the expressiveness and ease of understanding obtained with the current state-of-the-art single agent description within AI, i.e., a Bayesian network representation of the situation. Hence, the natural extension in order to

make the Bayesian game truly useful for other problems than leisure games is to maintain several influence diagram representations of the possible models and let the game's utility functions consist of the utilities that can be calculated with the use of the respective influence diagrams.

For a situation picture to be truly useful for a commander, it should convey both awareness of the current situation as well as predictive awareness regarding likely future courses of events. Hence, prediction of future courses of events must be considered of utmost importance when commencing development of the next generation's C2 systems and, henceforth, in higher level fusion the game component is both important and needed.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the help of Per Svensson at the Swedish Defence Research Agency for commenting on drafts at different stages.

REFERENCES

[1] D. S. Alberts
Network centric warfare: Current status and way ahead.
Journal of Defence Science, **8**, 3 (Sept. 2003), 117–119.

[2] D. S. Alberts, J. J. Garstka and F. P. Stein
Network Centric Warfare: Developing and Leveraging Information Superiority.
Washington, D.C.: CCRP, DoD, 2nd ed., 1999.

[3] S. Arnborg, H. Artman, J. Brynielsson and K. Wallenius
Information awareness in command and control: Precision, quality, utility.
In *Proceedings of the Third International Conference on Information Fusion* (FUSION 2000), Paris, France, July 2000, ThB1/25–32.

[4] R. Azoulay-Schwartz and S. Kraus
Negotiation on data allocation in multi-agent environments.
Autonomous Agents and Multi-Agent Systems, **5**, 2 (June 2002), 123–172.

[5] D. Billings, N. Burch, A. Davidson, R. Holte, J. Schaeffer, T. Schauenberg and D. Szafron
Approximating game-theoretic optimal strategies for full-scale poker.
In *Proceedings of the 2003 International Joint Conference on Artificial Intelligence* (IJCAI 2003), 2003, 661–668.

[6] J. Brynielsson
Using AI and games for decision support in command and control.
Decision Support Systems, in press.

[7] J. Brynielsson and S. Arnborg
Bayesian games for threat prediction and situation analysis.
In P. Svensson and J. Schubert (Eds.), *Proceedings of the Seventh International Conference on Information Fusion* (FUSION 2004), Stockholm, Sweden, vol. 2, June 28–July 1, 2004, 1125–1132.

[8] J. Brynielsson and S. Arnborg
Refinements of the command and control game component.
In *Proceedings of the Eighth International Conference on Information Fusion* (FUSION 2005), Philadelphia, PA, July 2005.

[9] J. Brynielsson, M. Engblom, R. Franzén, J. Nordh and L. Voigt
Enhanced situation awareness using random particles.
In *Proceedings of the Tenth International Command and Control Research and Technology Symposium* (ICCRTS), McLean, VA, June 2005.

[10] J. Brynielsson and K. Wallenius
Game environment for command and control operations (GECCO).
In *Proceedings of the First International Workshop on Cognitive Research With Microworlds*, Granada, Spain, Nov. 2001, 85–95.

[11] V. Chvátal
Linear Programming.
New York: W. H. Freeman and Company, 1983.

[12] T. P. Coakley
Command and Control for War and Peace.
Washington, D.C.: National Defense University Press, 1991.

[13] R. W. Cottle, J-S. Pang and R. E. Stone
The Linear Complementarity Problem.
Boston, MA: Academic Press, 1992.

[14] R. G. Cowell, A. P. Dawid, S. L. Lauritzen and D. J. Spiegelhalter
Probabilistic Networks and Expert Systems.
Statistics for Engineering and Information Science, New York: Springer-Verlag, 1999.

[15] G. Devetag and M. Warglien
Playing the wrong game: An experimental analysis of relational complexity and strategic misrepresentation.
CEEL Working Paper 0504, Computable and Experimental Economics Laboratory, Department of Economics, University of Trento, Italy, 2005.

[16] E. Durschmied
The Hinge Factor: How Chance and Stupidity Have Changed History.
London: Coronet Books, 1999.

[17] H. Friman and B. Brehmer
Using microworlds to study intuitive battle dynamics: A concept for the future.
In *Proceedings of Command and Control Research and Technology Symposium* (CCRTS), Newport, RI, June 29–July 1, 1999.

[18] Y. Gal and A. Pfeffer
A language for modeling agents' decision making processes in games.
In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems* (AAMAS'03), Melbourne, Australia, July 2003, 265–272.

[19] I. Gilboa and E. Zemel
Nash and correlated equilibria: Some complexity considerations.
Games and Economic Behavior, **1**, 1 (Mar. 1989), 80–93.

[20] P. J. Gmytrasiewicz and E. H. Durfee
Rational coordination in multi-agent environments.
Autonomous Agents and Multi-Agent Systems, **3**, 4 (Dec. 2000), 319–350.

[21] J. C. Harsanyi
Games with incomplete information played by "Bayesian" players.
Management Science, **14**, 3,5,7 (1967–1968), 159–182, 320–334, 486–502.

[22] G. E. Horne
Maneuver warfare distillations: Essence not verisimilitude.
In P. A. Farrington, H. B. Nembhard, D. T. Sturrock and G. W. Evans (Eds.), *Proceedings of the 1999 Winter Simulation Conference*, 1999.

[23] E. Horvitz, J. Breese, D. Heckerman, D. Hovel and K. Rommelse
The Lumière project: Bayesian user modeling for inferring the goals and needs of software users.
In G. F. Cooper and S. Moral (Eds.), *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence* (UAI-98), Madison, WI, July 1998, 256–265.

[24] R. A. Howard and J. E. Matheson
Influence diagrams.
In R. A. Howard and J. E. Matheson (Eds.), *Readings on the Principles and Applications of Decision Analysis*, Menlo Park, CA: Strategic Decisions Group, 1984, 721–762.

[25] R. Johansson and R. Suzić
Particle filter-based information acquisition for robust plan recognition.
In *Proceedings of the Eighth International Conference on Information Fusion* (FUSION 2005), Philadelphia, PA, July 2005.

[26] R. Johansson, N. Xiong and H. I. Christensen
A game theoretic model for management of mobile sensors.
In *Proceedings of the Sixth International Conference on Information Fusion* (FUSION 2003), Cairns, Australia, July 2003, 583–590.

[27] A. Katz and B. Butler
"Game commander"—Applying an architecture of game theory and tree lookahead to the command and control process.
In *Proceedings of the Fifth Annual Conference on AI, Simulation, and Planning in High Autonomy Systems* (AIS94), Gainesville, FL, 1994.

[28] G. A. Klein
Recognition-primed decisions.
In W. B. Rouse (Ed.), *Advances in Man-Machine Systems Research*, Greenwich, UK: JAI Press, 1989, 47–92.

[29] D. E. Knuth and R. W. Moore
An analysis of alpha-beta pruning.
Artificial Intelligence, **6**, 4 (Winter 1975), 293–326.

[30] D. Koller, N. Megiddo and B. von Stengel
Fast algorithms for finding randomized strategies in game trees.
In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing* (STOC), Montréal, Québec, Canada, May 1994, 750–759.

[31] D. Koller, N. Megiddo and B. von Stengel
Efficient computation of equilibria for extensive two-person games.
Games and Economic Behavior, **14**, 2 (June 1996), 247–259.

[32] D. Koller and B. Milch
Multi-agent influence diagrams for representing and solving games.
Games and Economic Behavior, **45**, 1 (Oct. 2003), 181–221.

[33] D. Koller and A. Pfeffer
Representations and solutions for game-theoretic problems.
Artificial Intelligence, **94**, 1 (July 1997), 167–215.

[34] D. M. Kreps
Game Theory and Economic Modelling.
Clarendon Lectures in Economics, Oxford University Press, 1990.

[35] J. Kuylenstierna, J. Rydmark and H. Sandström
Some research results obtained with DKE: A dynamic wargame for experiments.
In *Proceedings of the Ninth International Command and Control Research and Technology Symposium* (ICCRTS), Copenhagen, Denmark, Sept. 2004.

[36] C. E. Lemke and J. T. Howson, Jr.
Equilibrium points of bimatrix games.
Journal of the Society for Industrial and Applied Mathematics, **12**, 2, (June 1964), 413–423.

[37] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz and F. E. White, Jr.
Revisiting the JDL data fusion model II.
In P. Svensson and J. Schubert (Eds.), *Proceedings of the Seventh International Conference on Information Fusion* (FUSION 2004), Stockholm, Sweden, vol. 2, June 28–July 1, 2004, 1218–1230.

[38] J-F. Mertens and S. Zamir
Formulation of Bayesian analysis for games with incomplete information.
International Journal of Game Theory, **14**, 1 (Mar. 1985), 1–29.

[39] J. Moffat
Complexity Theory and Network Centric Warfare.
Washington, D.C.: CCRP, DoD, 2003.

[40] K. P. Murphy
Dynamic Bayesian Networks: Representation, Inference and Learning.
Ph.D. thesis, University of California, Berkeley, 2002.

[41] R. B. Myerson
Game Theory: Analysis of Conflict.
Cambridge, MA: Harvard University Press, 1991.

[42] J. F. Nash
Non-cooperative games.
Annals of Mathematics, **54**, 2 (Sept. 1951), 286–295.

[43] S. Parsons and M. Wooldridge
Game theory and decision theory in multi-agent systems.
Autonomous Agents and Multi-Agent Systems, **5**, 3 (Sept. 2002), 243–254.

[44] J. Pearl
Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.
San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1988.

[45] S. J. Russell and P. Norvig
Artifical Intelligence: A Modern Approach.
Upper Saddle River, NJ: Prentice Hall, 2nd ed., 2003.

[46] R. D. Shachter
Evaluating influence diagrams.
Operations Research, **34**, 6 (Nov.–Dec. 1986), 871–882.

[47] A. Steinberg, C. Bowman and F. E. White, Jr.
Revisions to the JDL data fusion model.
In *Proceedings of SPIE AeroSense* (Sensor Fusion: Architectures, Algorithms, and Applications III), Orlando, FL, vol. 3719, Apr. 1999, 430–441.

[48] B. von Stengel
Efficient computation of behavior strategies.
Games and Economic Behavior, **14**, 2 (June 1996), 220–246.

[49] B. von Stengel
Computing equilibria for two-person games.
In R. J. Aumann and S. Hart (Eds.), *Handbook of Game Theory with Economic Applications*, Elsevier Science, vol. 3 of Handbooks in Economics, 2002, 1723–1759.

[50] R. Suzić
Representation and recognition of uncertain enemy policies using statistical models.
In *Proceedings of the NATO RTO Symposium on Military Data and Information Fusion*, Prague, Czech Republic, Oct. 2003.

[51] P. Thunholm
Military Decision Making and Planning: Towards a New Prescriptive Model.
Ph.D. thesis, Department of Psychology, Stockholm University, Sweden, June 2003.

[52] U.S. Army
Intelligence Preparation of the Battlefield.
Field Manual FM 34–130, Headquarters, Dept. of the Army, Washington, D.C., July 1994.

[53] R. Vane and P. Lehner
Using hypergames to increase planned payoff and reduce risk.
Autonomous Agents and Multi-Agent Systems, **5**, 3 (Sept. 2002), 365–380.

[54] K. Virtanen, J. Karelahti and T. Raivio
Modeling air combat by a moving horizon influence diagram game.
Journal of Guidance, Control, and Dynamics, **29**, 5 (Sept.–Oct. 2006), 1080–1091.

[55] K. Wallenius
Support for situation awareness in command and control.
In P. Svensson and J. Schubert (Eds.), *Proceedings of the Seventh International Conference on Information Fusion* (FUSION 2004), Stockholm, Sweden, vol. 2, June 28–July 1, 2004, 1117–1124.

[56] F. E. White, Jr.
A model for data fusion.
In *Proceedings of the First National Symposium on Sensor Fusion*, Orlando, FL, vol. 2, Apr. 1988.

[57] N. Xiong and P. Svensson
Multi-sensor management for information fusion: Issues and approaches.
Information Fusion, **3**, 2 (June 2002), 163–186.

Joel Brynielsson was born in 1974 in Stockholm, Sweden, where he earned his M.Sc. degree in computer engineering and his Ph.D. degree in computer science, both from the Royal Institute of Technology. He is currently an assistant professor at the Swedish National Defence College.

His research interests are in information and uncertainty management in command and control systems with papers appearing in journals and conferences devoted to information fusion, decision support, command and control, operations research, microworld research, and modeling and simulation.

Stefan Arnborg took the civilingenjör (M.Sc.E.) exam in engineering physics at the Royal Institute of Technology (KTH) in 1968, and the Ph.D. in information processing at KTH in 1972.

He worked as a supercomputer programmer at the Courant Institute (1966), with object-oriented programming systems at the Norwegian Computing Centre (1969–1970) and CAP Sweden (1971), as an operations research analyst at the Swedish Defence Research Establishment (FOA, 1972–1979), and as a distributed systems and computer guru at Philips Financial Terminal Systems (PEAB, 1979–1982). He spent sabbatical terms at the University of Oregon (1986, 1989, and 1993) and was part-time advisor to the Swedish Institute of Computer Science (SICS). He is the director of the theory group and of the M.Sc.E. program in computer engineering. He has been a professor of computer science at KTH since 1982.

His research interests are algorithms and complexity, verification, computer algebra, data engineering and uncertainty management, data mining, human brain informatics, command and control, and aesthetics in engineering education.

Issues and Challenges in Situation Assessment (Level 2 Fusion)

ERIK BLASCH
IVAN KADAR
JOHN SALERNO
MIECZYSLAW M. KOKAR
SUBRATA DAS
GERALD M. POWELL
DANIEL D. CORKILL
ENRIQUE H. RUSPINI

Situation assessment (SA) involves deriving relations among entities, e.g., the aggregation of object states (i.e., classification and location). While SA has been recognized in the information fusion and human factors literature, there still exist open questions regarding knowledge representation and reasoning methods to afford SA. For instance, while lots of data is collected over a region of interest, how does this information get presented to an attention constrained user? The information overload can deteriorate cognitive reasoning so a pragmatic solution to knowledge representation is needed for effective and efficient situation understanding. In this paper, we present issues associated with Level 2 Information Fusion (Situation Assessment) including: (1) user perception and perceptual reasoning representation, (2) knowledge discovery process models, (3) procedural versus logical reasoning about relationships, (4) user-fusion interaction through performance metrics, and (5) syntactic and semantic representations. While a definitive conclusion is not the aim of the paper, many critical issues are proposed in order to characterize future successful strategies for knowledge representation, presentation, and reasoning for situation assessment.

Manuscript received May 8, 2006; revised Oct. 7, 2006; released for publication Nov. 28, 2006.

Refereeing of this contribution was handled by Dr. Chee-Yee Chong.

Authors' addresses: E. Blasch, AFRL, Dayton, OH; I. Kadar, Interlink Systems Sciences, Inc.; J. Salerno, AFRL, Rome, NY; M. M. Kokar, Northeastern University, Boston, MA; S. Das, Charles River Analytics, Cambridge, MA; G. M. Powell, U.S. Army RDECOM CERDEC I2WD, Ft. Monmouth, NJ; D. D. Corkill, Univ. of Massachusetts, Amherst, MA; E. H. Ruspini, Artificial Intelligence Center, SRI International, Menlo Mark, CA.

1. INTRODUCTION

Situation assessment (SA) is an important part of the information fusion (IF) process because it (1) is the purpose for the use of IF to synthesize the multitude of information, (2) provides an interface between the user and the automation, and (3) focuses data collection and management. Hall and Llinas (Table I) have listed a variety of techniques that need to be solved for SA to be viably implemented in real systems [15]. Since the late 1990s there has been few cumulative updates in the progress of SA and still there are remaining issues and challenges. During the FUSION05 conference, Ivan Kadar organized, moderated, and participated in a panel discussion with invited leading experts to elicit and summarize current issues and challenges in SA that need to be researched in the next decade.

1.1. Panel Participants, Topics, and Perspectives

This paper serves as a retrospective view of the panel discussion that was held in July 2005. In this format, we list our retrospective and annotated view of the panel information in a condensed (bulletized) format to make it easier for the reader to assimilate the general concepts. Due to space limitation, only a few key issues are expanded on in text format.

- Organizer: Ivan Kadar, Interlink Systems Sciences, Inc.
- Co-Organizers: Subrata Das, Charles River Analytics and Mieczyslaw M. Kokar, Northeastern University
- Moderators: Ivan Kadar, Interlink Systems Sciences, Inc. and James Llinas, SUNY at Buffalo
- July 26, 2005 FUSION 2005—The 8th International Conference on Information Fusion, July 25–28, Philadelphia, PA

PARTICIPANTS AND PRESENTATION TITLES

- "Knowledge Representation Issues in Perceptual Reasoning Managed Situation Assessment" Ivan Kadar, Interlink Systems Sciences, Inc., Lake Success, NY
- "Knowledge Representation Requirements for Situation Awareness" John Salerno, Douglas Boulware, Raymond Cardillo, Air Force Research Laboratory, Rome Research Site, NY
- "Situation Assessment: Procedural versus Logical" Mieczyslaw M. Kokar, Department of Elect. & Computer Eng., Northeastern University, Boston, MA
- "Tactical Situation Assessment Challenges and Implications for Computational Support" Gerald M. Powell, U.S. Army RDECOM CERDEC I2WD, Ft. Monmouth, NJ
- "Situation Assessment in Urban Combat Environments" Subrata Das, Charles River Analytics, Inc., Cambridge, MA

TABLE I
SA Challenges and Limitations—Hall and Llinas, [15]

JDL Process	Processing Description	Current Status	Challenges and Limitations
Level 2	Develops a description of current relationships among objects and events in the context of the environment (i.e., situation assessment)	Numerous prototypes Dominance by Knowledge-Based Systems (KBS) —Blackboard methods —Rule-based representation —Logical templates KBS experiments —Case based reasoning, Fuzzy Logic Non-real time implementation	Dominated by prototypes No experience on scaling to field models "Excedrin" cognitive models Difficult KB development Perfunctory Test & Evaluation Integration of identity/kinematic data

- "Representation and Contribution-Integration Challenges in Collaborative Situation Assessment" Daniel D. Corkill, University of Massachusetts, Amherst, MA
- "Human-Aided Multi-Sensor Fusion" Enrique H. Ruspini, et al., Artificial Intelligence Center, SRI International, Menlo Mark, CA
- "DFIG Level 5 (User Refinement) issues supporting Level 2 (Situation Assessment)" Erik Blasch, AFRL, WPAFB, OH

1.2. Common Themes

While discussion of individual research results by the participants highlighted specific key issues, there were common themes that resulted from the panel discussion. The common themes were:

COMMON ISSUES

- User—The SA process includes perceptual, interactive, and human control
- Process models—updating behavioral models (e.g. Bayes Nets, procedural/logical, perceptual, learning)
- Context—operational situation (i.e., dependent on the current state of the environment)
- Meaning—semantics and syntax issues (formal methods, ontologies)
- Metrics—develop a standard set of metrics (e.g. trust, bounds, uncertainty)

COMMON CHALLENGES

- Explanation of process—evidence accumulation and contradiction in knowledge representation and reasoning
- Graphical displays to facilitate inferential chains, collaborative interaction, and knowledge presentation
- Interactive control for corrections and utility assessment for knowledge management

2. SITUATION AWARENESS/SITUATION ASSESSMENT

There are two main communities that are looking at situational information (i.e., Situation Awareness (SAW) and Situation Assessment (SA)): the human factors community and the engineering information fusion (IF) research community. SAW is a mental state while SA supports (e.g. fusion products) that state. The human factors notion of SAW is being lead by Mica Endsley [12]. For the IF society, there are many leading people proposing different aspects of SAW research. Research is a way to categorize developments, but another way is by applications. There are many application communities looking at SAW research including: military, medical, aviation, security, and environmental. Each might have differences, but the commonality rests in the fact that a multitude of data needs to be synthesized into a single operating picture (dimensionality reduction) [37]. Likewise, the salient information needs to be provided to the user to assist the user in completing their mission tasks.

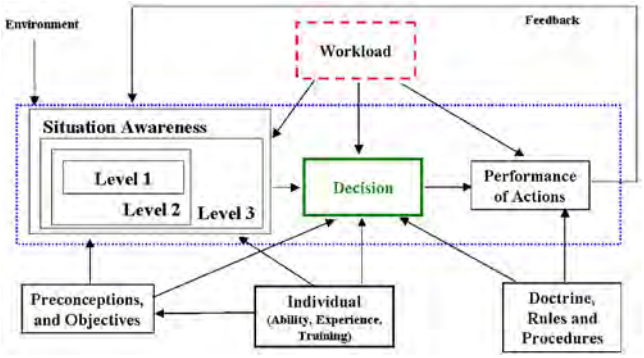


Fig. 1. Endsley's situation awareness model.

2.1. Situational Awareness Models

The Human in the Loop (HIL) of a semi-automated system must be given adequate situation awareness. According to Endsley "SAW is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future." [12]. This now-classic model, shown in Fig. 1, translates into 3 levels:

- Level 1 SAW—Perception of elements in the environment

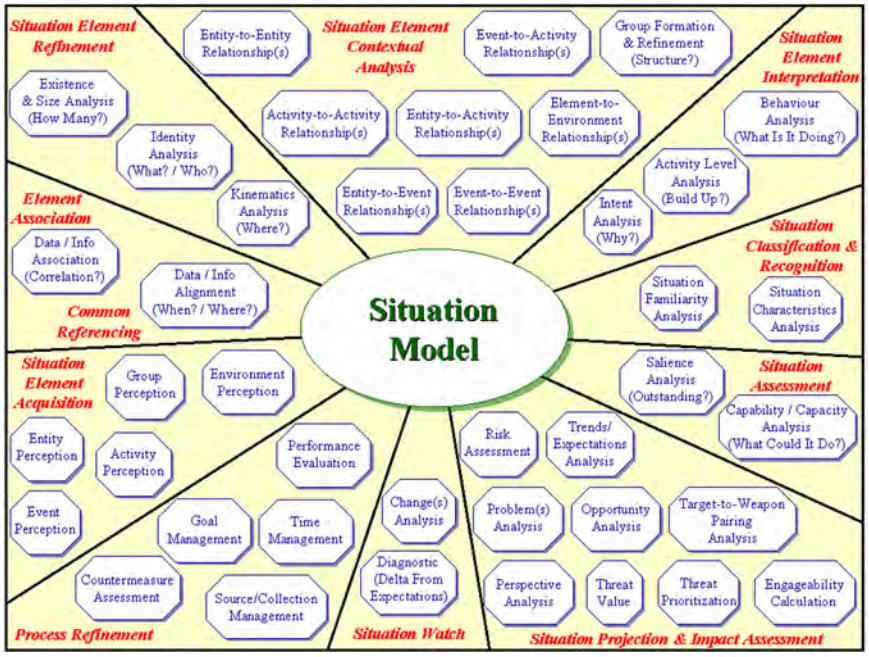


Fig. 2. Fusion situation awareness model [4].

- Level 2 SAW—Comprehension of the current situation
- Level 3 SAW—Projection of future states

Operators of dynamic systems use their SAW in determining their actions. To optimize decision making, the SAW provided by an IF system should be as precise as possible as to the objects in the environment (Level 1 SAW). A SA approach should present a fused representation of the data (Level 2 SAW) and provide support for the operator's projection needs (Level 3 SAW) in order to facilitate the operator's goals. From the SA model presented in Fig. 1, workload is a key component of the model that affects not only SAW, but also the decision and reaction time of the user.

2.2. User Fusion Model

As another example, the Situational Model components [32], shown in Fig. 2, developed by Roy, show the various information needs to provide the user with an appropriate SAW. To develop the SA model further, we note that the user must be primed for situations to be able to operate faster, and more effectively.

A fusion system must satisfy the user's functional needs and extend their sensory capabilities. Of interest to the information fusion community are IF systems which translate data about a region of interest into knowledge, or at least information over which the human can reason and make decisions. A user fuses data and information over time and space and acts through their world mental model—whether it be in the head or with graphical displays, tools, and techniques. The current paradigm for fusion research, shown in Fig. 3, is called the user-fusion model [5].

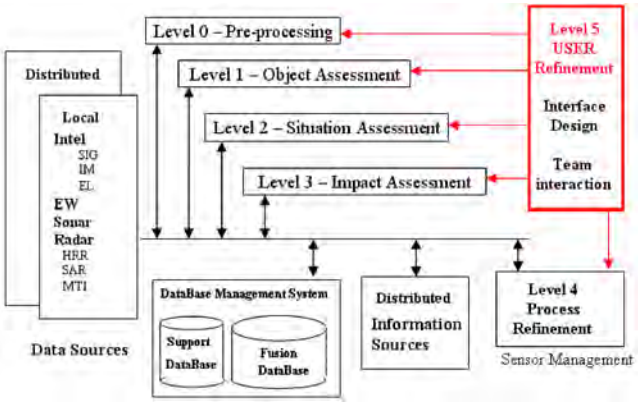


Fig. 3. User fusion model.

2.3. Perceptual Reasoning Managed Situation Assessment

"Knowledge Representation Issues in Perceptual Reasoning Managed Situation Assessment" Ivan Kadar

The IF community has had several definitions of SA over time. The JDL Model [14], defined SA as "estimation and prediction of relations among entities, to include force structure and cross force relations, communications and perceptual influences, physical context, etc." DSTO [11, 22] defined SA as "an iterative process of fusing the spatial and temporal relationships between entities to group them together and form an abstracted interpretation of the patterns in the order of battle data." Issues with the SA definitions, and some subsequent models based on these definitions are:

- not domain independent,
- do not incorporate human thought processes, human perceptual reasoning, the ability to control sensing and essence of response time,

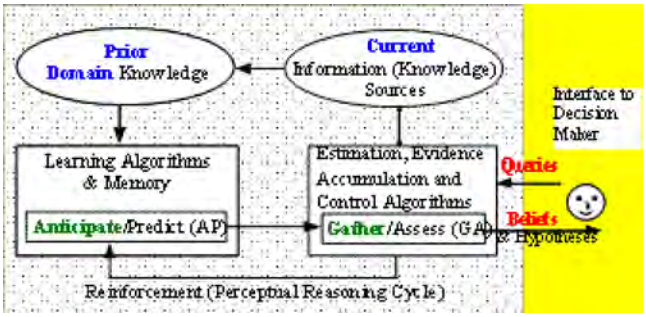


Fig. 4. Perceptual reasoning machine.

- imply use of limited a priori information,
- and only imply potential for new knowledge capture.

Therefore, the desired properties of SA are:

- One needs the ability to control Levels 1–4 of Data Fusion processes for knowledge capture in SA
- SA is to establish relationships (not necessarily hierarchical) and associations among entities, it should anticipate with a priori knowledge in order to rapidly gather, assess, interpret and predict what these relationships might be; it should plan, predict, anticipate again with updated knowledge, adaptively learn, and control the fusion processes for optimum knowledge capture and decision making
- These features are similar to the characteristics of human perceptual reasoning
- Therefore it is conjectured that the "optimum" SA system should emulate human thinking as much as possible

As a matter of fact, the godfather of the Internet and knowledge representation, Vannevar Bush [8] in his famous 1945 essay, "As We May Think" stated, op. cit., "The human mind does not work that way hierarchically. It operates by association." Spatial and temporal associations are key ingredients of Perceptual Reasoning Model (PRM).

The goal is the perceptual reasoning model which is viewed as a "meta-level information management system," as shown in Fig. 4. PRM consists of a feedback planning/resource control system whose interacting elements are: "assess," "anticipate" and "predict" [16–18].

- Gather/Assess current, Anticipate future (hypotheses), and Predict information requirements and monitor intent,
- Plan the allocation of information/sensor/system resources and acquisition of data through the control of a separate distributed multisource sensors/systems resource manager (SRM),
- Interpret and act on acquired (sensor, spatial and contextual) data in light of the overall situation by interpreting conflicting/misleading information.

Representative elements and knowledge bases, associated with the assess, anticipate and predict PRM modules, are categorizable into: (1) functions, with each function further categorized into (a) knowledge required, (b) knowledge acquisition methods, (c) knowledge representation approaches, and (d) implementation techniques. Specific knowledge representation and reasoning (KRR) methods were discussed at the panel highlighting implementation issues and research challenges.

Issues for SA
1. Knowledge—a priori and current
2. PROCESS—anticipate and gather facts
3. User queries instantiation
4. Fusion System presents Beliefs
5. Need a process model interface

KRR Challenges for SA
1. Adequacy of KRR (logic, ontology, algorithmic, probabilistic), how to quantify/measure?
2. Expressiveness of models versus tractability of inference
3. Managing Complexity (how to bound problem w/incomplete knowledge)
4. Data Information (How to manage heterogeneous and uncertain KSs and detect duplicate or incomplete concepts)
5. Presentation of knowledge to different users (what is pragmatic?)

2.4. Syntactic Algorithms and Semantic Synonyms

"Knowledge Representation Requirements for Situation Awareness" John Salerno, Doug Boulware, Ray Cardillo

Full Spectrum Dominance (FSD), as defined by Joint Vision 2020, is the ability to be persuasive in peace, decisive in war and preeminent in any form of conflict. FSD cannot be accomplished without the capability to know what the adversary is currently doing as well as the capacity to correctly anticipate the adversary's future actions. This ability of projection is an element of Situation Awareness [12, 13]. SA has received increased attention due to its diverse applications in a number of problem domains including: asymmetric threat, tactical, cyber, and homeland security [14]. Salerno, et al. proposes an architecture that combines the Endsley and JDL models (shown in Fig. 5) and has applied this model to various strategic, cyber and tactical applications [35].

Through a display, a user can (1) build a model by either editing an existing template/model or create a new one; (2) activate/de-activate existing models; or (3) view active models and any evidence that has been associated with the model over time. Different political, military, economic, social, infrastructure, and information models can be accessed and the result published (or subscribed) to.

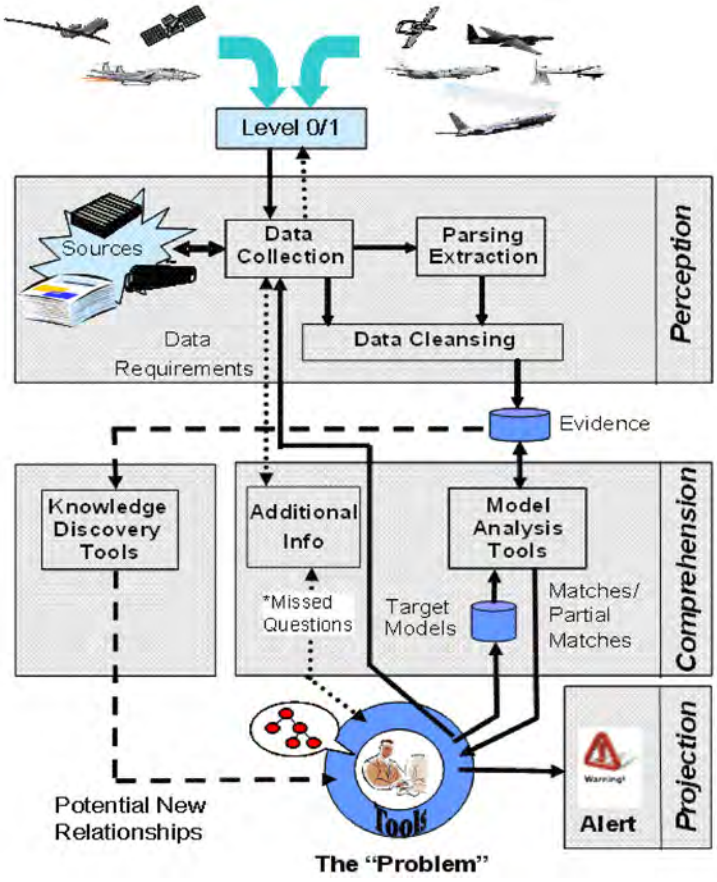


Fig. 5. SA framework.

Issues encountered in its development mainly pertain to evidence access, storage, usage and providing a priori knowledge. In order to resolve any semantic issues in context and value, we need to normalize the data before we can use it. Data normalization involves converting different formats of the same data into a common representation. Dealing with semantic inconsistencies is much more difficult. In these cases, we need to resolve synonyms both in what is represented and what the value itself represents. Two different labels can have the same meaning, or two aliases can represent the same entity. Finally, what level of a priori data is needed depends on the context of operation.

Issues for SA
1. Lots of data for analyst, but not able to get it
2. Analyst—under stress and fatigue
3. What to publish and subscribe
4. Security issues in data gathering

Challenges for SA
1. Syntactic algorithms (normalization/transformation)
2. Semantic synonyms (different meaning between ideas)
3. Learning from what is presented
4. People can think of new situations

2.5. Procedural versus Logical

"Situation Assessment: Procedural versus Logical" Mieczyslaw M. Kokar

Various terms have been used to refer to Level 2 fusion processing: situation refinement, situation awareness, situation development, relation estimation and other. All of these terms have a common part in their definition, i.e., all of them require that the definition should include the knowledge of all the relevant objects and their kinematic states. This is essentially a Level 1 function, so it will not be discussed here. Some of the definitions, but not all, include the requirement of knowing relationships among the objects. This brings three problems: 1) The relevance problem: there are so many possible relations—which ones are relevant? 2) The resource problem: where can we get the necessary information resources, both data and processing, that can be used to assess the current situation? 3) The derivation problem: how do we derive whether a particular relation holds or not? And even fewer definitions capture the aspect of awareness as defined in the Webster dictionary, where awareness is explained as "AWARE implies vigilance in observing or alertness in drawing inferences from what one experiences." In other words, a subject is aware if the subject not only observes (experiences) the objects but also is capable of drawing conclusions from

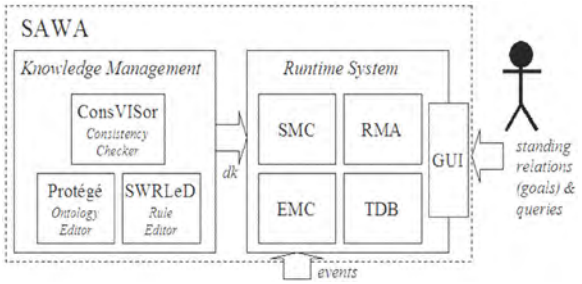


Fig. 6. SAWA. Situation management component (SMC), relation monitor agent (RMA), triples data base (TDB), and event management concept (EMC).

these observations. We call this the inference problem: how can we infer the implications of a specific situation on the tasks that we are pursuing?

The observations presented in this paper have been collected during the two year period of working on the situation awareness assistant (SAWA), shown in Fig. 6 [27]. In most general terms, SAWA is an ontology based situation monitor [26]. Its main goal is to monitor a "standing relation," i.e., a query formulated in terms of an underlying ontology. SAWA collects information (events) and invokes its inference engine that derives whether the relation holds or not. The reasoning mechanism of SAWA combines logical inference with Bayesian belief propagation. A number of findings from this project have been published in papers [20, 21, 26, 27, 28].

Solutions are sought by either procedural or logical (declarative) means. In the logical approach, a query about a specific relation can be posed to an inference engine (or a theorem prover). The inference engine then returns an answer, possibly with some variable bindings. A number of inference engines for OWL have been developed and/or are under development. In typical data fusion applications the derivation problem is solved in a procedural way, i.e., in order to determine whether a particular relation holds or not, a procedure is invoked, which returns either a "yes" or a "no" answer, possibly also including some return parameters. While this approach may turn out to be more efficient in terms of time complexity, it lacks the genericity that the logical approach has. The limitation comes from the fact that only those queries for which procedures have been explicitly coded by the system developer can be answered. The logical approach is termed declarative programming, while the procedural approach is called procedural programming.

In the logical approach, the inference problem is closely related to the relation derivation problem. A logical query regarding any feature of a situation is posed to an inference engine. The query language for OWL is called OWL-QL. The number of types of queries is only limited by the complexity of the ontology that captures the domain knowledge. The queries are built out of the class expressions and property expressions using logical connectives that are part of the ontology language. Again, the advantage of the logical approach is that the query engine is not designed to answer a specific set of queries, but it is rather generic, capable of answering any query that is expressible in the query language. This is not the case in the procedural approach, where only those queries that have been formulated at the design time can be resolved by the system. The reasoning mechanism of SAWA combines logical inference with Bayesian belief propagation. Although the logical approach is a promising approach to solve the general SA problem, still, a number of issues need to be resolved in order to make the logical approach scalable up to the real world problems.

Issues for SA
Relations—Future in Semantic Web approach 1. Relevance—need a generic relevance theory 2. Resource—from closed (level 1 provides all information) to open (level 2 accesses Semantic Web for additional knowledge) 3. Derivation/Inference—expressiveness versus efficiency of reasoning
Challenges for SA
1. Consistency and ontology mapping 2. Identity crisis (association problem) 3. Representational expressiveness, computational complexity 4. Trust Metrics 5. "Semantic Web"—use standard language (i.e. "OWL"), but need more expressiveness (rules)

3. USERS AND APPLICATIONS

3.1. Tactical SA and Computational Support

"Tactical Situation Assessment Challenges and Implications for Computational Support" Gerald M. Powell

A number of definitions of Level 2 fusion are available [2, 3, 23]. The comments in this paper relate to one or more of these definitions. The operational focus is Army brigade intelligence analysis. There exist approaches to instantiate these definitions into practical designs [3]. Fig. 7 shows the representational information from Waltz [36] which shows the context-dependent perceptual knowledge views for processing. These displays show spatial and temporal relations from which to act. The display technology is domain dependent and requires operational considerations [30–31]. What follows is a small subset of key issues and challenges in knowledge representation and reasoning methods for Level 2 fusion in this task domain.

Hypotheses and Their Utilities: There is a need to generate hypotheses to serve as predictors of behavior, to guide information gathering, and to provide a framework for constructing plausible explanations for evidence. The goal of creating hypothesis structures that

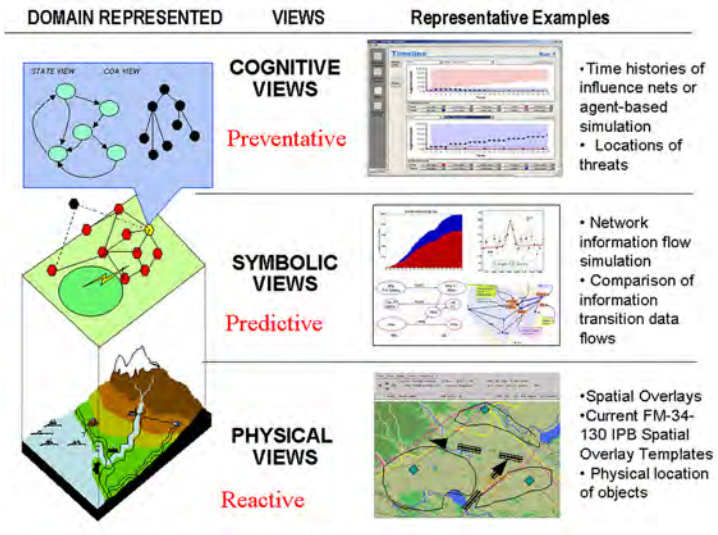


Fig. 7. Categories of view.

will satisfy these purposes would indicate an adequate understanding exists of the hypothesis types required and the logical relationships among them. The relative importance of a given hypothesis in the structure will be context-dependent—this requires analysis. Similarly, the relative value of reports/data from differing source types and instances will be context-dependent. Like hypotheses, their relevance or value, will vary over time as the situation unfolds including what information has already arrived, whether it has been analyzed, the quality of it, its relative importance and so on. A report deemed irrelevant in a particular context may, much later, become relevant as the interpretation has evolved. Analysis and interpretation are context-dependent. They must take place within the set of context-dependencies defined by the information peculiar to a given situation (mission, terrain, battlespace reports, etc.) as well as historical knowledge about the adversary. These dependencies can cause combinatoric growth in the number of interpretations possible and lead to erroneous analyses and conclusions. Identifying what these dependencies are, and constructing ways to represent and reason with them to produce increased accuracy and speed of analysis and interpretation remain open issues.

Weak Models of the Adversary: In some situations, our knowledge of the actors we are observing and trying to understand may be extremely weak such as when there has been little opportunity for information gathering prior to engagement, when their organizational elements and communications patterns are partitioned in ways that inhibits discovery of structure, and when their doctrine and tactics encourage rapid, adaptive changes in behavior sometimes manifesting in unexpected ways. Even when opportunities for observation are plentiful, accurately interpreting data in a timely manner may be extremely difficult due to indicators that are weak discriminators of hypotheses. These issues indicate there are implications for both directed and undirected machine-based knowledge discovery. Also, models and tools to help analysts understand situational-specific risks of Type I (false positive) and II (false-negative) errors in interpretation would be useful.

Multiple Inferencing Strategies: Abductive, deductive and inductive inferencing are present in human performance in situation assessment. There are implications for machine capabilities to support each of these in an integrated framework. Their machine implementations should be such that they support user understanding, trust and acceptance.

Issues for SA
1. Massive information overload on analysts
2. Analysis and interpretation are context-dependent
3. Cognitive biases cause errors in analysis and interpretation
4. Models of adversary structure/behavior are often weak
5. Heterogeneous, non-integrated information sources
6. Automated environments supporting adequately fast, direct authoring of knowledge by analysts do not exist

Challenges for SA
1. Automated analysis/interpretation that is fast enough while also being accurate
2. Overcoming representational and processing complexities caused by context-dependence
3. System designs that will help analysts overcome cognitive biases
4. Automatic adaptation to changing threats
5. Semantic consistency across info sources
6. Building adequate knowledge authoring environments for analysts

3.2. Urban Combat

"Situation Assessment in Urban Combat Environments" Subrata Das

The two largest hurdles for SA in contemporary urban combat environments are the environmental clut-

ter and the enemy's lack of conformity to established tactical doctrine. Adversarial entities in the environment must be identified and tracked, individually or as groups, to recognize higher-level situations (e.g. attack, ambush, interdiction, insurgency) and determine effective military responses or preemptive actions. Furthermore, because contemporary enemy behavior is often innovative and unpredictable, traditional tactical models cannot be applied to recognize significant developments in contemporary situations. As a result, an effective automated means for extracting useful situation information from the thousands of multi-source events generated every minute in the theatre of operations remains an open problem. Human analysts currently perform the bulk of this difficult situation and threat assessment work, but are only able to process a small fraction of the available data. Knowledge discovery (aka. data mining) is a process of abstracting knowledge from data to form models for problem solving. Knowledge discovery techniques such as Decision Trees and Inductive Logic Programming discover association rules between items within an unordered collection of records, transactions, or events. Techniques also exist for extracting causal Bayesian belief network (BN) structures along with their strengths. BN technology offers several advantages, including its easy-to-understand graphical modeling and consistent probabilistic semantics in dealing with the uncertainty involved in sensor data. Focusing now on the model-based approach, current state-of-the-art approaches for answering the commander's priority intelligence requirements (PIRs) for SA are model-based. Knowledge discovery or model-based approaches fail to provide a complete solution for SA requirements because a) they can only model specific patterns within a relatively small subset of voluminous data, and b) there is never enough historical data available to model novel phenomena. To address these issues, we explore a hybrid approach combining model-based reasoning with knowledge discovery techniques for SA, especially suitable for detecting and identifying asymmetric threats in urban environments. The proposed hybrid approach leverages the wealth of data available to provide information about "what is strange" about a given situation, without having to know what exactly what it is we are looking for, thus triggering models for follow-up SA.

The hybrid approach recognizes significant patterns by taking into account environmental clutter. It also uses spatiotemporal clustering algorithms to perform a space and time-series analysis of messages without requiring semantic information. This approach can, for example, detect spatially correlated moving units over time within the environment. Detected patterns trigger follow-up assessment of newly developed situations, resulting in invocations of various doctrine-based computational models, including causal static and dynamic Bayesian belief networks. The invoked models then perform SA based on other observables propagated as evidence into the models. The approach extends further in recognizing significant patterns without relying on doctrinal knowledge. Instead, we make use of latent semantic indexing (LSI), which is a proven technique in text based information retrieval applications. We leverage LSI to extract underlying patterns from observables reported in formatted (e.g. USMTF) or plain text messages. These patterns establish a "normal" profile against which subsequent incoming observations are matched so as to detect any unusual activities (e.g. large scale attack preparation).

Issues for SA
1. Model Based Reasoning—closed form of reasoning and model construction process is time consuming
2. Traditional Knowledge Discovery—requires large amount of training data
3. Link Analysis—manual process and not able to handle large amount of data
Challenges for SA
1. Rapid construction of models
2. BN—for model building (all pair-wise interactions)
3. Unsupervised clustering techniques for large volumes of data to generate normalcy and determine "something is going on"
4. Automation to find "needle in a haystack"

3.3. Collaborative Situation Awareness

"Representation and Contribution-Integration Challenges in Collaborative Situation Assessment" Daniel D. Corkill

Blackboard systems are an ideal architecture for situation assessment involving large data volumes and heterogeneous data and knowledge sources. However, the ad hoc confidence and belief values used in traditional blackboard applications have led to criticism of the blackboard approach and spawned efforts to combine collaborative blackboard-system techniques with more "principled" graphical-network representations. We discuss two important collaborative-assessment challenge areas: 1) principled blackboard representations and 2) principled integration of contributions made by independent knowledge-source entities. The complexity of these challenges is highlighted using a simple assessment scenario, shown in Fig. 8(a).

The effectiveness of blackboard systems is the product of a number of architectural capabilities working in concert. The first important capability is the control flexibility provided by indirect, anonymous, and temporally disjoint interaction among software entities. The blackboard-system control shell can delay execution of a knowledge-source (KS) execution without having to modify an explicit process or worry about managing the data needed by the delayed KS—they remain on the blackboard. Similarly, KS activations can be exe-

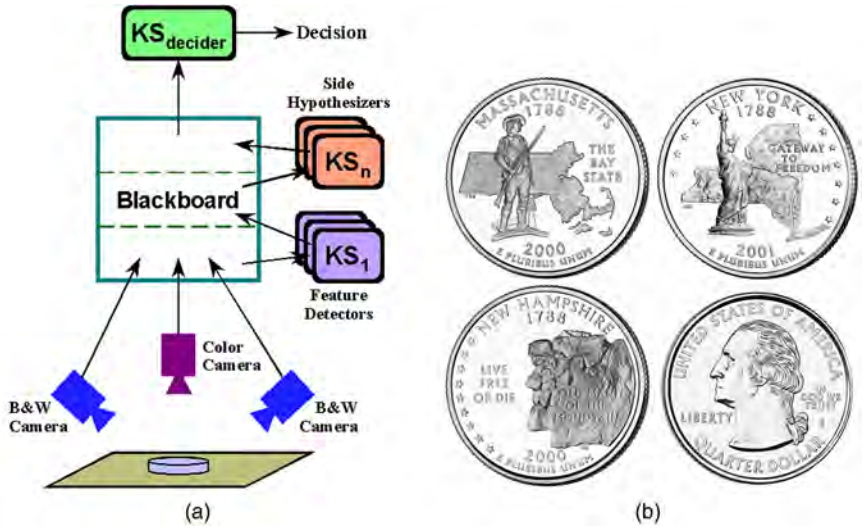


Fig. 8. Fair coin detector.

cuted earlier than normal—whenever there appears to be sufficient information for them to perform useful work. Preliminary efforts in applying graphical belief networks to blackboard systems have focused on a principled representation of the developing solution on the blackboard [34]. Current beliefs are represented on the blackboard as disconnected graphical network [9, 10, 29]. The emphasis should be on making the integration of the contributions made by diverse entities well founded. This can only be achieved by modeling how these contributions are generated and how they relate to one another. For example, if two KSs use the same data and produce similar results using different computational approaches, how independent are the results? Are they redundant (with no added certainty in the results) or complementary (in the sense that each has the potential to make mistakes on certain data values, but these mistakes are fully independent of one another)?

The Fair-Coin Problem: To illustrate these challenges, consider a simple collaborative-assessment problem of deciding if a U.S. quarter is a fair coin (has a head and a tail) by observing a series of coin flips. A priori we are told that there is a 50% chance that the quarter is either two-headed or two-tailed. We have a tabletop that can be viewed by three cameras: two black-and-white cameras and a color camera (Fig. 8(a)). Images feed into our assessment architecture that includes a number of KSs. There are low-level KSs that attempt to identify coin features, higher-level KSs that aggregate features to hypothesize coin sides, and a decider KS that makes the fair or non-fair-coin designation. The goal is to make a principled determination with a specific confidence with as few flip observations as possible. Adding to the complexity is the U.S. 50 State Quarter program, where a new quarter with a state-specific reverse side is issued every 10 weeks in the order that the states were admitted into the Union (Fig. 8(b)).

Issues for SA
1. Blackboard architectures Different knowledge sources Benefit from shared information Bayesian blackboard systems Graphical belief nets (procedural) 2. Integration of contribution systems

Challenges for SA
1. Representation of uncertainty and certainty 2. Develop entity-specific behavioral specifications of contributions 3. Specifications provided by user for computer to learn 4. Development of feature-identification Knowledge-source 5. Use of characteristics in concert 6. How to deal with mistakes in condition characterizations

3.4. Human Aided Situation Awareness

"Human-Aided Multi-Sensor Fusion" Enrique H. Ruspini, Artificial Intelligence Center, SRI International

In multi-sensor fusion problems, relevant knowledge cannot be completely represented by computer models. In these cases, it is necessary to implement mechanisms that permit human experts to apply the full range of knowledge that only they can master. We identify two fundamental requirements for such a system. It is first necessary to identify properties of a reasoning system that may be visualized by humans so as to judge the credibility and reliability of its results. In addition, it is necessary to implement control and review procedures that may be applied by humans to improve fusion results. We believe that any sophisticated human-aided multi-sensor system that addresses these two needs must provide the following capabilities:

a) Knowledge acquisition procedures
b) Explicit representation of multi-sensor knowledge

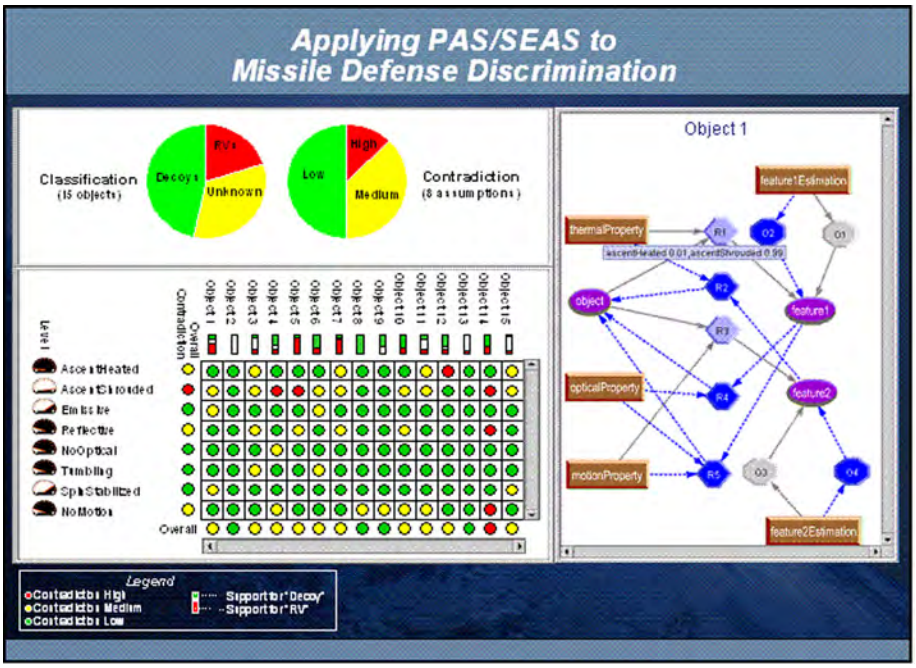


Fig. 9. Structural evidential argumentation system.

c) Quantitative indicators of properties of fusion results

d) Intuitive, understandable displays of those properties

e) Interactive techniques to improve the quality of fusion results

We propose a human-aided multi-sensor fusion system based on the integration of the Probabilistic Argumentation System (PAS) [4], developed by Lockheed Martin, and the Structural Evidential Argumentation System (SEAS) [25], developed by SRI International, shown in Fig. 9. These two software tools implement variants of the Dempster-Shafer (DS) calculus of evidence [19]. PAS is a formalism that explicitly encodes assumptions by means of logical rules in the context of a generalized probabilistic framework. The reasoning procedures of PAS produce measures of support and plausibility for various conclusions while also providing mechanisms to explain the nature of the inferential chains employed to arrive at those results. SEAS permits the recording of analytical processes employed by intelligence analysts to derive their findings. SEAS was originally developed to support collaborative reasoning among multiple analysts. SEAS provides intuitive graphical displays that enable analysts to review analytical processes, their underlying assumptions, and the nature of the processes employed to arrive at conclusions. In practice, the structured-argumentation processes employed by SEAS have been shown to facilitate quick understanding of analytical processes while permitting capture of the collective thinking of groups of analysts.

The integration of PAS and SEAS attempts to satisfy the previously requirements by developing:

a) Logical rules to facilitate the acquisition and explicit representation of knowledge

b) DS calculus of evidence to provide a powerful mechanism to model sensor evidence and uncertain knowledge

c) Explanations about the fusion processes to permit quantification of the relevance of various knowledge items and the detection and identification of contradictions while enabling consideration of alternative hypotheses

d) Graphical displays to facilitate understanding of inferential chains and their conclusions

e) Interactive control and review mechanisms to permit humans to correct arguments to increase the utility of conclusion and fusion results

Issues for SA
1. Knowledge acquisition systems
2. Explicit representation of multi-sensor knowledge
3. Quantitative indicators of properties of results
4. Intuitive, understandable displays of those properties
5. Interactive techniques to improve fusion results quality

Challenges for SA
1. Logical rules to facilitate acquisition
2. DS—evidence for uncertain knowledge
3. Explanation of process—evidence and contradiction
4. Graphical displays to facilitate inferential chains
5. Interactive control for corrections and utility of conclusions

3.5. User Refinement—Level 5 of DFIG Model

"DFIG Level 5 (User Refinement) issues supporting Level 2 (Situation Assessment)." Erik Blasch

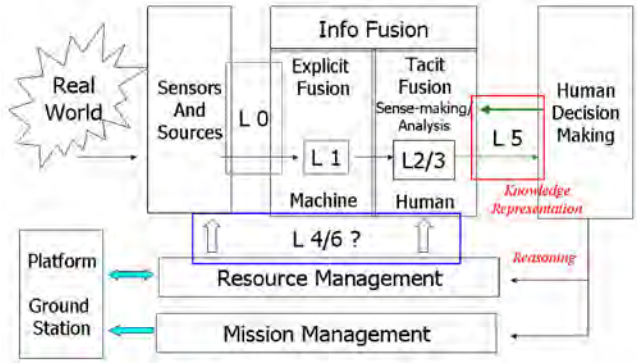


Fig. 10. DFIG 2004 model.

The current fusion model supporting the evaluation and deployment of sensor fusion systems is the User-Fusion model, [7], shown in Fig. 3, with upgrades from the current Data Fusion Information Group[1] (DFIG) (which is the current JDL). The key for SA is the user's mental model [1]. The mental model is the representation of the world as aggregated through the data gathering, IF design, and the user's perception of the social, political, and military situations.

The DFIG model, shown in Fig. 10, separates the data fusion and management functions. Management functions are divided into sensor control, platform placement, and user selection to meet mission objectives. Level 2 (SA) includes tacit functions which are inferred from level 1 explicit representations of object assessment. Since the unobserved aspects of the SA problem can not be processed by a computer, user knowledge and reasoning is necessary. The current definitions, based on the revised JDL fusion model [7], include: (see for other revisions [24])

Level 0—Data Assessment: estimation and prediction of signal/object observable states on the basis of pixel/signal level data association (e.g. information systems collections);

Level 1—Object Assessment: estimation and prediction of entity states on the basis of data association, continuous state estimation and discrete state estimation (e.g. data processing);

Level 2—Situation Assessment: estimation and prediction of relations among entities, to include force structure and force relations, communications, etc. (e.g. information processing);

Level 3—Impact Assessment: estimation and prediction of effects on situations of planned or estimated actions by the participants; to include interactions between action plans of multiple players (e.g. assessing threat actions to planned actions and mission requirements, performance evaluation);

Level 4—Process Refinement (an element of Resource Management): adaptive data acquisition and processing to support sensing objectives (e.g. sensor management and information systems dissemination, command/control).

Level 5—User Refinement (an element of Knowledge Management): adaptive determination of who queries information and who has access to information (e.g. information operations) and adaptive data retrieved and displayed to support cognitive decision making and actions (e.g. human computer interface).

Level 6—Mission Management (an element of Platform Management): adaptive determination of spatial-temporal control of assets (e.g. airspace operations) and route planning and goal determination to support team decision making and actions (e.g. theater operations) over social, economic, and political constraints.

For SA, the user must (1) prioritize information needs to the fusion manager, (2) require reliable and validated information, and (3) seek patterns [6]. The information priority is based on the information desired. The user must have the ability to choose or select the objects of interest and the processes from which the raw data is converted to the fused data. One of the issues in the processing of fused information is related to ability to understand the information origin or pedigree. It is important to note that reliability and validity are two different concepts. A piece of information can be 100% reliable and either totally diagnostic (100% validity) or un-diagnostic (0% validity) in predicting information. However, the less reliable the information, the less valid it is because of the inherent uncertainty (i.e., error) in the information itself.

Users have individual differences for Reasoning Methods (RM) and thus, the coordination between the user and the machine needs to be flexible. An example is that one user might look at sensor data while another might plan missions (see Fig. 10). The responsibility of the user thus determines the information needs requirements for SA. To be able to facilitate many users, a control strategy needs to be defined wherein the user can query and update the database. One way to facilitate user opportunities, a standard set of metrics for Knowledge Representation (KR) should be designed that afford Quality. Blasch [6] explored the concepts of level 2, situation awareness or assessment, by detailing the user needs of attention, workload, and trust which can be mapped into metrics of timeliness, throughput, confidence, and accuracy. Table II lists metrics for SAW as referenced to the communications, human factors, automatic target recognition (ATR), and target tracking literature. SA is hard to define and creates interface problems if not standardized. Information needs of fusion systems for KR and RM need rigorous testing in experimental designs to define SA Products. Additionally, dynamic updating of Knowledge Delivery for planning requires timely and reliable data for reasoning.

[1]Frank White, Otto Kessler, James Llinas, Alan Steinberg, Dave Hall, Ed Waltz, Gerald Powell, Mike Hinman, John Salerno, Erik Blasch, Dale Walsh, Chris Bowman, Mitch Kokar, Joe Karalowski, Richard Antony.

TABLE II
Metrics for Fusion and Situational Awareness

COMM	Human Factors	Sit Aware*	ATR	TRACK
Delay	Reaction Time	Timeliness	Acquisition/Run Time	Update Rate
Probability of Error	Confidence	Confidence	Prob. (Hit), Prob. (FA)	Probability of Detection
Delay Variation	Attention	Purity, Precision	Positional Accuracy	Covariance
Throughput	Workload	Usage	# Images	No. Targets
Cost	Cost	Utility	Collection platforms	No. Assets
Security	Trust	Reliabilty	Ontology, Taxonomy	Cooperative Nav.

*Tadda et al. propose some of these for Cyber SA: purity for quality detection, evidence recall, and attack score [35].

Issues for SA
1. Standard Set of Metrics for Knowledge Representation
2. User (individual differences) for Reasoning Methods
3. Dynamic updating of Knowledge Delivery for Planning
4. Users desire a variety of SA display information
5. Information Needs of fusion systems for KR and RM

Challenges for SA
1. Scoping a common terminology and metrics
2. Affording control strategies for different users
3. KR must afford timeliness for reasoning
4. Interface design must be flexible (KR) to different users
5. Rigorous testing in experimental designs to define SA

4. CONCLUSIONS

The panel discussion highlighted many different, but common themes that are SA issues and proposed a variety of challenges of SA for the future. The common issues are: (1) User focused (perceptual, interactive, control), (2) developing Process models for behavioral modeling and updating the models (e.g.—Bayes Nets, procedural/logical, perceptual, learning), (3) determining the Context—operational situation (i.e. domain dependent), (4) detailing the Meaning (i.e. semantics and syntactic relations), and (4) the need for a standard set of SA Metrics (e.g. trust, bounds, uncertainty). The common challenges include (a) explanation of process that addresses evidence accumulation and contradiction constraints for knowledge representation and reasoning, (b) graphical displays to facilitate inferential chains, collaborative interaction, and knowledge representation, and (c) interactive control for corrections and utility assessment for knowledge management. While these lists are notional, the information presented is from a panel of participants who have all tried to build SA tools for the operator and thus, the issues and challenges are posed from experience. The next phase of the collaboration research on SA design, issues, and challenges will focus on a set of process models. Possible directions and extensions include utilization of intelligent agents to emulate team cognition [38], use of gaming concepts for hypothesis generation and data understanding, and rapid evolution of human-computer interaction such as 3-D full immersion environments.

REFERENCES

[1] S. Andriole and L. Adelman
Cognitive Systems Eng. for User-Computer Interface Design, Prototyping, & Evaluation.
Lawrence Erlbaum, 1995.

[2] R. Antony
Principles of Data Fusion.
Boston-London: Artech House, 1995.

[3] K. Baclawski, M. Kokar, J. Letkowski, C. Matheus, and M. Malczewski
Formalization of situation awareness.
In *Proceedings of Eleventh OOPSLA Workshop on Behavioral Semantics*, 2002, 1–15.

[4] T. Baker, M. Chan, and T. Hansen
A Java implementation of the PAS for data fusion in missile defense applications.
In *Proceedings of SPIE*, Vol. 5434, April 14–15, 2004, 176–186.

[5] E. Blasch
Assembling an information-fused human-computer cognitive decision making tool.
IEEE Aerospace and Electronic Systems Magazine, (June 2000), 11–17.

[6] E. Blasch
Situation, impact, and user refinement.
SPIE Aerosense, **5096** (2003), 270–279.

[7] E. Blasch and S. B. Plano
JDL level 5 fusion model "user refinement" issues and applications in group tracking.
SPIE Aerosense, **4729** (2002), 270–279.

[8] V. Bush
As we may think.
The Atlantic Monthly, July 1945.

[9] D. D. Corkill
Blackboard systems.
AI Expert, **6**, 9 (Sept. 1991), 40–47.

[10] D. D. Corkill
Collaborating software: Blackboard and multi-agent systems & the future.
In *Proceedings of the International Lisp Conference*, New York, Oct. 2003.

[11] DSTO (Defence Science and Technology Organization)
Data Fusion Special Interest Group, Data fusion lexicon.
Department of Defence, Australia, Sept. 21, 1994, 7p.

[12] M. R. Endsley
Design and evaluation for situation awareness enhancement.
In *Proceedings of the Human Factors Society 32nd Annual Meeting* Santa Monica, CA, Human Factors and Ergonomics Society, 1988, 97–101.

[13] M. R. Endsley
Toward a theory of situation awareness in dynamic systems.
Human Factors Journal, **37**, 1 (Mar. 1995), 32–64.

[14] [JDL]
Data fusion lexicon.
U.S. DOD, Data Fusion Subpanel of the Joint Directors of Laboratories, Tech. Panel, 1991.

[15] D. L. Hall and J. Llinas
Introduction to multisensor data fusion.
Proceedings of IEEE, **85**, 1 (Jan. 1997), 6–23.

[16] I. Kadar
Data fusion by perceptual reasoning and prediction.
In *Proceedings 1987 Tri-Service Data Fusion Symposium*, Johns Hopkins University Applied Physics Laboratory, Laurel, MD, June 1987.

[17] I. Kadar
Perceptual reasoning managed situation assessment and adaptive fusion processing.
In Ivan Kadar (Ed.), *Proceedings of Signal; Processing, Sensor Fusion & Target Recognition X*, *Proceedings of SPIE*, Vol. 4380, Apr. 2001, 385–396.

[18] I. Kadar
Perceptual reasoning in adaptive fusion processing.
In Ivan Kadar (Ed.), *Proceedings Signal; Processing, Sensor Fusion and Target Recognition XI*, *Proceedings of SPIE*, Vol. 4729, Apr. 2002, 342–351.

[19] J. Kohlas and P. A. Monney
A Mathematical Theory of Hints. An Approach to the Dempster-Shafer Theory of Evidence.
Lecture Notes in Economics and Mathematical Systems Series, New York: Springer Verlag, 1995, 425.

[20] M. M. Kokar, C. J. Matheus, K. Baclawski, J. A. Letkowski, M. Hinman, and J. Salerno
Use cases for ontologies in information fusion.
In *Proceedings of the 7th International Conf. on Information Fusion*, 2004, 415–421.

[21] M. M. Kokar, C. J. Matheus, K. Baclawski, J. A. Letkowski, M. Hinman, and J. Salerno
Association in Level 2 fusion.
In *Proceedings of SPIE International Soc. Opt. Eng.* Vol. 5434, 2004, 228–237.

[22] D. A. Lambert
Situations for situation awareness.
In *Proceedings of Fusion 2001*, 2001.

[23] J. Llinas
On the scientific foundations of Level 2 fusion.
Presentation at *The RTO IST Symposium on Military Data and Information Fusion*, Prague, Czech Republic, Oct. 2003.

[24] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz, and F. White
Revisions and extensions to the JDL data fusion model II.
In *Proceedings of the 7th International Conference on Information Fusion* (Fusion 2004), Stockholm, Sweden, June 2004, 1218–1230.

[25] J. D. Lowrance, I. W. Harrison, and A. C. Rodriguez
Capturing analytic thought.
In *Proceedings of 1st International Conference on Knowledge*, Capture, Oct. 2001, 84–91.

[26] C. J. Matheus, K. P. Baclawski, and M. M. Kokar
Derivation of ontological relations using formal methods in a situation awareness scenario.
In *Proceedings of SPIE International Soc. Opt. Eng.*, 2003, 298–309.

[27] C. J. Matheus, M. M. Kokar, et al.
SAWA: An assistant for higher-level fusion and situation awareness.
SPIE, Vol. 5813, 2005, 75–85.

[28] C. J Matheus, M. M. Kokar, and K. Baclawski
A core ontology for situational awareness.
In *Proceedings of the 6th International Conference on Information Fusion* (Fusion 2003), 2003, 545–552.

[29] J. Pearl
Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.
Morgan Kaufmann, 1988.

[30] G. Powell and B. Broome
Fusion based knowledge for the objective force.
In *Proceedings of NSSDF*, San Diego, CA, 2002.

[31] G. Powell and C. Brown, III
User-centered identification and analysis of operational requirements for upper levels of fusion at the unit of action.
In *Proceedings of NSSDF*, JHU, APL, Baltimore, MD, June 2004.

[32] J. Roy
From data fusion to situation analysis.
In *Proceedings of the 4th International Conference on Information Fusion* (Fusion 2001), Montreal, Aug. 7–10, 2001, Vol. 1, TuD1-16 to TuD1-23.

[33] J. J. Salerno, M. Hinman, and D. Boulware
Building a framework for situation awareness.
In *Proceedings of the 7th International Conference on Information Fusion* (Fusion 2004), Stockholm, Sweden, June 2004, 219–226.

[34] C. Sutton, C. Morrison, P. R. Cohen, J. Moody, and J. Abidi
A Bayesian blackboard for information fusion.
In *Proceedings of the 7th International Conference on Information Fusion* (Fusion 2004), Stockholm, Sweden, June 2004, 1111–1116.

[35] G. Tadda, J. J. Salerno, D. Boulware, M. Hinman, and S. Gorton
Realizing situation awareness within a cyber environment.
In *Proceedings of SPIE*, Vol. 6242, 2006.

[36] E. Waltz
Data fusion in offensive and defensive information operations.
NSSDF Symposium, 2000.

[37] E. Waltz and J. Llinas
Multisensor and Data Fusion.
Boston, MA: Artech House, 1990.

[38] J. Yen, X. Fan, and R. A. Volz
Information-needs in agent teamwork.
Web Intelligence and Agent Systems: An International Journal, **2**, 4 (2004), 231–247.

Erik Blasch received his B.S. in mechanical engineering from the Massachusetts Institute of Technology and Masters in mechanical, health science, and industrial engineering from Georgia Tech and attended the University of Wisconsin for an M.D./Ph.D. in Mech. Eng./Neurosciences until being called to active duty in the United States Air Force. He completed an M.B.A., M.S.E.E., M.S. Econ. M.S./Ph.D. psychology (ABD), and a Ph.D. in electrical engineering from Wright State University (WSU). He is currently still a student in Air Staff College.

Dr. Blasch is an information fusion evaluation tech lead for the Air Force Research Laboratory—COMprehensive Performance Assessment of Sensor Exploitation (COMPASE) Center, adjunct EE and BME professor in at WSU and AFIT, and a reserve Major with the Air Force Office of Scientific Research. He was a founding member of the International Society of Information Fusion (ISIF) in 1998 and is the 2007 ISIF President. Dr. Blasch has many military and civilian career awards; but engineering highlights include team member of the winning '91 American Tour del Sol solar car competition, '94 AIAA mobile robotics contest, and the '93 Aerial Unmanned Vehicle competition where they were first in the world to automatically control a helicopter. Since that time, Dr. Blasch has focused on automatic target recognition, targeting tracking, and information fusion research compiling 180 scientific papers and book chapters. He is active in IEEE (AES and SMC) and SPIE including regional activities, conference boards, journal reviews, and scholarship committees; and participates in the Data Fusion Information Group (DFIG), the Program Committee for the DOD Automatic Target Recognition Working Group (ATRWG), and the SENSIAC National Symposium on Sensor and Data Fusion (NSSDF).

Ivan Kadar received the B.E.E. degree from City College of CUNY, the M.S.E.E. degree from Columbia University and the Ph.D. degree in electrical engineering from Polytechnic Institute of New York.

In 2000 he founded Interlink Systems Sciences, Inc. providing scientific and technical consulting services in systems, sensors and algorithms with applications to all levels of multi-source information/sensor/data fusion and related technologies, and currently consults part-time to private industry and universities. In addition, he is also currently employed at Northrop Grumman Corporation Integrated Systems, as Associate Fellow, and serves as technical advisor in the broad area of Information Fusion Technologies. Previously, he worked for Grumman/Northrop Grumman Corporation in the roles of consultant, principal scientist, technical advisor, principal engineer and technical specialist responsible for conceptual design, technical direction, original applied research/advanced development, organization and supervision of professionals, management of R&D programs and integration of advanced technologies. He initiated, managed and/or was principal investigator on hands-on research/development in all levels of information/sensor/data fusion, tracking, automated target recognition, knowledge-based intelligent systems, communications, and systems integration for 19 years. He also worked for IBM T. J. Watson Research Center as a research staff member in the Systems Analysis, Algorithms, Networking and Satellite Communications Group of Computer Sciences. His experience/research areas of interests include systems design and synthesis; geometrical, connection and resource management aspects of centralized, distributed and adaptive information fusion and tracking, and its interactions with higher-level fusion processing; feature-aided multi-sensor multi-target tracking and robust association; perceptual reasoning and anticipation in situation assessment and intent modeling; uncertainty models; evidential reasoning; knowledge representation and management; long-term prediction of target states using contextual knowledge; sensor modeling; fusion systems measures-of-merits, optimization and performance predictions; robust estimation with applications to tracking and sensor data processing; pre-and-post-detection fusion; statistical image processing; target/pattern recognition; ad-hoc sensor networks; communications and radio navigation systems; distributed decision making; artificial intelligence; decision aiding; and medical informatics with applications to real-world problems.

Dr. Kadar has authored/coauthored over 100 papers, book chapters, edited two books and fifteen volumes of the *Signal Processing, Sensor Fusion and Target Recognition* conference which he organizes and chairs; chair for Sensor Data Exploitation and Target Recognition Conferences of the SPIE Defense and Security Symposia; serves on the program and/or technical committees for the above, and for some of the International Conferences on Information Fusion; elected to the editorial board of the on-line journal *Advances in Information Fusion* of ISIF, and served on the editorial board of the *International Journal of Information Fusion*. He is recipient of the IEEE Region I award, and two IEEE AES M. Barry Carlton Best Paper Awards. He is reviewer for several journals. He was elected in 2005 to serve on the Board of Directors of the International Society of Information Fusion (ISIF) (2006–2008); and elected as SPIE Fellow in 2007.

John Salerno received an A.A.S. degree in mathematics from Mohawk Valley Community College, a B.A. in math and physics from SUNY at Potsdam, a M.S.E.E. from Syracuse University and a Ph.D. in computer science from SUNY Binghamton. His research interests include neural networks and natural language. He began his career with the Air Force Laboratory in 1980 as a computer scientist in the Communications Networks Branch. The nature of his assignments at AFRL has included positions as research scientist, team leader, branch chief and assistant division chief. Over his outstanding career, he has effectively honed and applied his unique skills to solve complex operational Air Force and Intelligence Community problems. In doing so, Dr. Salerno has established a prodigious record of designing and successfully transitioning innovative solutions from early in-house feasibility demonstrations to fielded operational systems, many of which are supporting customers worldwide today.

Dr. Salerno has authored over 40 publications and 10 DoD technical publications, has pioneered 3 Air Force Inventions (patents pending), and has consistently proven his ability to organize and advance the state-of-the-art in Information Fusion technology. John's reputation for excellence has earned him advisory positions with 4 international conferences as well as invited membership with the DoD International Collaboration on Communications; OSD Science and Technology Panel on Data Fusion, and the DoD Assessment of Redeployment of Personnel to the Persian Gulf. As a technical leader, he has personally developed and guided multiple teams to deliver seminal operational capabilities such as Project Broadsword, HyperText Markup Language (HTML) Interface for Imagery Product Library (IPL), and the Client Server Environment. He envisioned, organized, and built a unique Intelligence/Air Force capability, the Joint Integration Test Facility (JITF), to perform initial operational evaluation on critical intelligence software systems. He currently leads the world-recognized Fusion 2+ group to develop and integrate technologies from across AFRL and other DoD research laboratories to deliver a groundbreaking warfighter capability to collect, interpret, and anticipate adversarial actions/intent. He has been married for the past 21 years to Barbara and has three children; Steven, Eric and Nicole.

Mieczyslaw M. Kokar has an M.S. and a Ph.D. in computer systems engineering from Wroclaw University of Technology, Poland.

Dr. Kokar is with the Department of Electrical and Computer Engineering at Northeastern University in Boston. His technical research interests include information fusion, ontology-based information processing, self-controlling software and modeling languages. In particular, he is interested in higher-level information fusion and situation awareness, ontology-based software radios, the specification and design of self-controlling software using the control theory metaphor, ontology development, ontological annotation of information, logical reasoning about OWL annotated information, consistency checking, formalization of the UML language, consistency checking of UML models versus UML Metamodel and of UML Metamodel versus MOF. He teaches various graduate courses in software engineering, formal methods and artificial intelligence. He is a senior member of the IEEE and member of the ACM.

More information about Professor Kokar can be found at his web site: http://www.ece.neu.edu/groups/scs/kokar.

Subrata Das received his Ph.D. in computer science from Heriot-Watt University in Scotland and a M.Tech. from the Indian Statistical Institute.

He is the chief scientist at Charles River Analytics, Inc. (www.cra.com) in Cambridge, MA. Subrata leads research projects in the areas of high-level and distributed information fusion, decision-making under uncertainty, intelligent agents, planning and scheduling, and machine learning. His technical expertise includes mathematical logics, probabilistic reasoning including Bayesian belief networks, symbolic argumentation, particle filtering, and a broad range of computational artificial intelligence techniques. Subrata held research positions at Imperial College and Queen Mary and Westfield College, both part of the University of London.

He has published many journal and conference articles. He is the author of the book entitled *Deductive Databases and Logic Programming,* published by Addison-Wesley, and has coauthored the book entitled *Safe and Sound: Artificial Intelligence in Hazardous Applications,* published by the MIT Press. His forthcoming book entitled, *Foundations of Decision Making Agents: Logic, Modality, and Probability,* is due shortly for publication by the World Scientific/Imperial College Press.

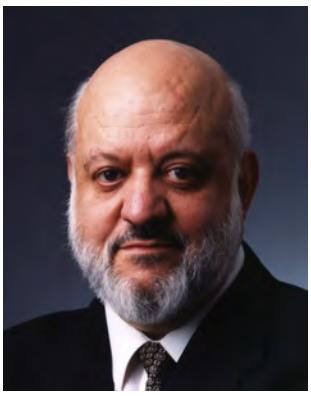
Dr. Subrata is a member of the editorial board of the journal, *Information Fusion,* published by Elsevier Science. He is in the process of editing a special issue for the journal relating to agent-based information fusion. Subrata has been a regular contributor, a technical committee/invited panel member, and a tutorial lecturer at each of the last three International Conferences on Information Fusion. He has been invited to join the technical committee at the forthcoming fusion conference to be held in Quebec City in July 2007, and to be part of the guest panel that will discuss lessons learned from level 2 fusion system implementations. He has also been giving a series of tutorials on multi-sensor data fusion on behalf of the Technology Training Corporation.

Gerald Powell is a senior scientist in the Intelligence and Information Warfare Directorate at U.S. Army RDECOM, CERDEC, Fort Monmouth, NJ. His research focuses on computational approaches to intelligence analysis and information gathering.

Dr. Powell is a senior member of the IEEE and a member of the American Association for Artificial Intelligence.

Daniel Corkill received the Ph.D. in computer science from the University of Massachusetts Amherst in 1983.

Dr. Corkill is a senior research scientist and associate director of the Multi-Agent Systems Laboratory in the Department of Computer Science at the University of Massachusetts Amherst. He is a leading authority in the areas of blackboard systems, multi-agent organizations, and collaborating software technologies. Corkill founded and served as President of Blackboard Technology Group (BBTech), which specialized in commercial blackboard-system software (GBB) and collaborating software research. At BBTech, he helped develop GBB applications in many domains, including the Radarsat-I mission control system. GBB received Object Magazine's Editors Choice award for software in 1996. Prior to founding BBTech, Corkill was an associate research professor in computer science at the University of Massachusetts Amherst. He is currently leading development of blackboard-based systems for the U.S. Air Force, DARPA, and DND Canada. He also heads the GBBopen Project (http://GBBopen.org/), which is providing high-performance, blackboard-system and collaborating-agent technology as open source.



Enrique H. Ruspini received his degree of Licenciado en Ciencias Matemáticas from the University of Buenos Aires, Argentina, and his doctoral degree in system science from the University of California at Los Angeles.

He is a principal scientist with the Artificial Intelligence Center, SRI International (formerly Stanford Research Institute), which he joined in 1984. Prior to joining SRI, he held positions at the University of Buenos Aires, the University of Southern California, UCLA's Brain Research Institute, and Hewlett-Packard Laboratories.

He is one the earliest contributors to the development of fuzzy-set theory and its applications, having introduced its use to the treatment of numerical classification and clustering problems. He has also made significant contributions to the understanding of the foundations of fuzzy logic and approximate-reasoning methods. His recent research has focused on the application of approximate-reasoning techniques to the development of systems for intelligent control of teams of autonomous robots, information retrieval, information fusion, qualitative description of complex objects, intelligent data analysis, and knowledge discovery/pattern matching in large databases.

He has lectured extensively in the United States and abroad, is a Fellow of the Institute of Electrical and Electronic Engineers, a First Fellow of the International Fuzzy Systems Association, a Fulbright Scholar, and a SRI Institute Fellow. He was the general chairman of the Second IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'93) and of the 1993 IEEE International Conference on Neural Networks (ICNN'93). He is one of the founding members of the North American Fuzzy Information Processing Society and a recipient of that society's King-Sun Fu Award. He is a former member of the IEEE Board of Directors (Division X Director, 2003–2004), the past-president (President-2001) of the IEEE Neural Networks Council, and its past Vice-President of Conferences. He has led numerous IEEE technical, educational, and organizational activities, is also a member of the Administrative Committee of the IEEE Computational Intelligence Society.

Dr. Ruspini is the Editor in Chief (together with P. P. Bonissone and W. Pedrycz) of the *Handbook of Fuzzy Computation*, a member of the Advisory Board of the *IEEE Transactions on Fuzzy Systems*, the *International Journal of Fuzzy Systems*, and the *Journal of Advanced Computational Intelligence and Intelligent Informatics*, and a member of the editorial board of the *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems*, *Fuzzy Sets and Systems*, and *Mathware and Soft Computing*.

Hierarchical Track Association and Fusion for a Networked Surveillance System

J. ARETA
University of Connecticut

Y. BAR-SHALOM
University of Connecticut

M. LEVEDAHL
Raytheon Co.

K. R. PATTIPATI
University of Connecticut

In this paper we present a benchmark problem for data association based on a real-world networked surveillance system, and compare the behavior of several multidimensional assignment (MDA) algorithms. The problem consists of a set of N_s observers which transmit track/event reports to a fusion center through a particular (real-world based) communication network among one of N_n networks. The network discards the observer's track identity (ID), replacing it by a network-generated ID and the observer ID, thus losing the information on the origin of the tracks sent by each observer. The solution approach developed in this paper consists of a hierarchical decomposition of the problem. This hierarchical approach first eliminates the redundancy introduced by the communication network by using an MDA algorithm per each observer present, and then using another MDA algorithm to choose which 'non-redundant' reports to fuse. This decomposition drastically reduces the dimensionality of the problem from $N_s \times N_n$ to N_s problems of dimension N_n and one of dimension N_s.

A comparison of two association criteria, *normalized distance squared* (NDS) and *likelihood ratio* (LR), is carried out. *It is shown that the LR yields significantly superior results.* Also the selection of certain parameters in the likelihood ratio is discussed. Finally, to evaluate their performance, three different MDA algorithms are used in this setup, Lagrangean Relaxation based MDA, Sequential m-best 2D and Linear Programming. A thorough comparison of these algorithms in terms of the quality of their solutions as well as their run times is done showing some pitfalls and advantages of each.

Manuscript received October 25, 2006; revised July 03, 2007; released for publication August 9, 2007.

Refereeing of this contribution was handled by P. Singer.

Authors' addresses: J. Areta, Y. Bar-Shalom and K. Pattipati, Dept. of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269-2157, E-mail: (areta ybs krishna@engr.uconn.edu); M. Levedahl, Raytheon Co., McLean, VA 22101, E-mail: (mlevedahl @raytheon.com).

1. INTRODUCTION

Data association is a major component of surveillance systems where several sensors and/or several targets are present. The origin uncertainty of reports from each of the sensors makes it critical for the fusion center to operate properly, especially in scenarios with closely-spaced targets. In such scenarios there is no clear-cut evidence whether a new report belongs to a previous track or not. The work in this paper deals with the formulation and the solution of a particular multidimensional assignment (MDA) problem using three different association algorithms for performance comparison.

The problem presented here corresponds to a real world surveillance network system for missile launch events. In it, a set of sources provide "event" (track) estimates via a number of communication networks to a Fusion Center (FC) which has to perform data association prior to fusion. This network model provides a realistic setup—a benchmark problem—that allows for a proper evaluation of the algorithms proposed to solve it. This differs from previous work [20], where such algorithms have been tested on randomly generated costs, which do not necessarily show the real performance of the algorithms in practical settings. The track generation model is a simplified one, without process noise.

The problem involves a set of N_s sources (observers) that provide event estimates (reports/tracks) of an unknown number of launches from their own observations. These reports are transmitted via N_n communication networks to a Fusion Center that has to perform data association prior to fusion. The network used to transmit each report is randomly chosen every time a new report is ready to be sent. The parameters estimated by the observers (and to be fused at the FC) are the launch time, launch point coordinates and heading. Figure 1 shows a possible scenario where 4 observers report through 3 networks to the fusion center.

A particular feature of the network model is that the information needed to distinguish among reports from the same source transmitted through different networks is not available at the FC: the track identity (ID) assigned by the source is not passed on, only a track ID assigned by the network and the source ID accompany the track. This makes it necessary to detect track duplications among the messages with the same source ID that arrive on different networks. This duplication elimination is performed based on the association criterion between tracks from the same origin, according to the recently developed general Likelihood Ratio (LR) approach in [3]. Also the selection of certain parameters in the LR is discussed. Out of the tracks deemed to be from the same event from the same source, the one with the smallest uncertainty is kept.

The resulting data, rearranged into sensor lists, is then associated using the likelihood ratio criterion from [3] with any of the proposed multidimensional assignment methods. The tracks obtained after association

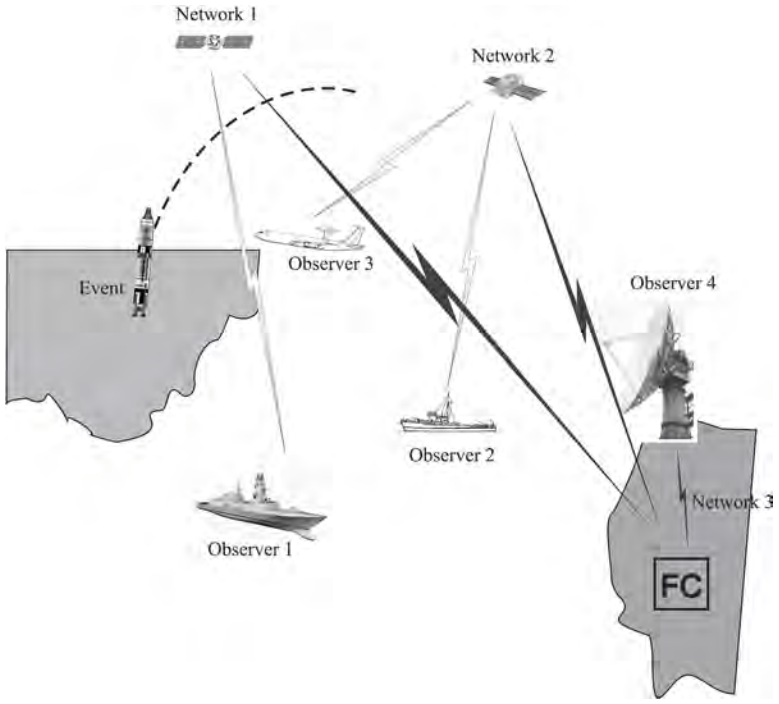


Fig. 1. Hypothetical scenario, where an event is reported by four observers to the FC using three communication networks.

are fused using a maximum likelihood (ML) approach as in [9]. An additional complication is that false reports can be also transmitted by the sources. Examples with several launches, sources and networks are presented to compare the performances of three assignment algorithms—the Lagrangean Relaxation based multidimensional (S-D) assignment [18], the "Sequential m-best 2D" assignment and the Linear Programming based assignment—on this realistic problem. The simpler Normalized Distance Squared (NDS) criterion is considered as well.

Section 2 describes the overall system with its layers and the information communicated between the layers, as well as a generic model for the local (source/observer) track/event estimates which are sent on the communication networks to the FC. Section 3 presents a hierarchical decomposition of the problem so that MDA algorithms can be applied in two stages. Section 4 presents two association criteria, the LR criterion, as well as the simpler normalized distance squared, also known as the Mahalanobis Distance or Chi-square criterion. Section 5 describes the three proposed methods to solve the MDA problem. Section 6 presents the results from the simulations and their analysis. Finally, Section 7 presents a discussion of the results and conclusions.

List of acronyms:

CC: Completely Correct
CI: Completely Incorrect
FC: Fusion Center
LaR: Lagrangean Relaxation
LR: Likelihood Ratio
LP: Linear Programming
MDA: Multidimensional (S-D) Assignment
ML: Maximum Likelihood
NDS: Normalized Distance Squared (Mahalanobis distance)
PC: Partially Correct
S-D: S-Dimensional Assignment
Sm2D: Sequential m-best 2D Assignment

2. PROBLEM FORMULATION

2.1. The Overall System

The system considered consists of the following layers:

1. Sources
2. Communication Networks
3. Fusion Center.

Each observer generates track reports and sends them to the FC through independent networks. These reports are based on individual observations made by the observers. Each event estimate consists of a vector and standard deviation (s.d.) for each component.[1] The reports received by the FC consist of

1. The event ID, t, which is *assigned by the network*. The source-assigned ID is not transmitted by the network to the FC, but replaced by a network-assigned

[1]The procedure easily generalizes to the case where one has full covariance matrices associated with the estimates, as long as they are available.

ID.[2] Nevertheless, the network-given ID continues to be used for all subsequent reports of the same event from the same observer.

2. The ID of the network, $n = 1,\ldots,N_n$.

3. The ID of the source, $s = 1,\ldots,N_s$.

4. The vector estimate and standard deviation for each component, as sent by the source.

The FC can receive reports on the same event from the same source via different networks—since the event ID is assigned by the network, it is not obvious upon reception at the FC which reports with different n and same s pertain to the same event/track. Also it is possible that reports with different n and different s can be on the same event. This makes it necessary to implement a data redundancy elimination stage prior to fusion, in order to eliminate duplicate reports on the same targets generated by the same observer.

2.2. The Sources

Each track report from a source is based upon all the measurements received up to the current time for that event. The measurements have uncorrelated errors and each report is the average of all the measurements received so reports on the same event have correlated errors.

It will be assumed that the measurement errors are zero mean and white with s.d. denoted as σ. While this value is not needed, it allows us to obtain the correlation coefficient of reports on the same event at the same source based on different (unknown) numbers of measurements. This is done as follows.

For the purpose of the present study, the estimate of an event based on k measurements is an n_x-vector with components assumed to be given by the (rather simple) expression[3]

$$\hat{x}_i(k) = \frac{1}{k}\sum_{l=1}^{k} z_i(l) \quad \text{with variance}$$
$$\sigma_i(k)^2 \triangleq E[(\hat{x}_i(k) - x_i)^2] = \frac{\sigma_i^2}{k}, \qquad i = 1,\ldots,n_x \tag{1}$$

where x_i is the true value of component i and the measurements are

$$z_i(l) = x_i + w_i(l), \qquad i = 1,\ldots,n_x \tag{2}$$

with the noises $w_i(l)$ zero mean, white and with variance σ_i^2. For the n_x components the noises are assumed to be uncorrelated. Note that the simple model (1) implies that there is no process noise and, consequently, no cross-correlation between the track estimates of the same event by different observers.

Then, given two estimates $\hat{x}_i(k_1)$ and $\hat{x}_i(k_2)$ from the same source, their correlation coefficient is [1]

$$\rho_i(t_1,t_2) = \frac{\min(\sigma_i(k_1)^{-2}, \sigma_i(k_2)^{-2})}{\sigma_i(k_1)^{-1}\sigma_i(k_2)^{-1}} \tag{3}$$

under the common origin assumption. This result will be used when carrying out certain track to track associations with correlated errors.

Note that estimates pertaining to the same event but obtained by different observers/sources have uncorrelated errors because there is no "common process noise" [4] since the above model (1) has no process noise at all.

On the other hand, if two tracks have different origins (i.e., they represent different events), their errors are uncorrelated. However, if one has two tracks with the same variance, the use of (3), which assumes common origin, would lead to unity correlation and the difference between the estimates—which is used in the test (see [4])—would then have zero variance. To avoid this, an upper bound (of, say, 0.95) could be used for (3).

We will consider false tracks but not tracks corrupted by false measurements.

3. HIERARCHICAL DECOMPOSITION OF THE PROBLEM

As discussed in Section 2, each report is accompanied by 3 indices: t (event/track number, assigned by the network), n (network number) and s (source number). In order to use an assignment algorithm for track to track association (to be followed by fusion) across several lists, where each list will be a collection of track/event reports that arrived at the FC via the various networks from a particular source, one has to make sure no track/event appears more than once in each list. Consequently, before the final association and fusion, one has to eliminate redundant tracks.

Figure 2 shows a schematic of the system, where the observers may transmit through any of the N_n communication networks. The fusion center explicitly shows its two main components, the *Removal of redundant tracks from the same source that arrived through different networks* and the *Track association across the N_s source lists* followed by fusion of the selected tracks.

The following operations are performed at the FC:

0. Removal of redundant tracks from the same source that arrived on the same network.

If one has two tracks from network n with the same t (and s), then only the most recent one (with the smallest variances) is kept. This eliminates "old (redundant) tracks" communicated through the same network that have been superseded by their updated versions.

1. Removal of redundant tracks from the same source that arrived through different networks.

[2]This peculiarity of the network, while strange from the researcher's point of view, is a real-world fact. Even if the network would have transmitted the observer's track ID, the problem would still be challenging.

[3]More general estimates can be used as long as the corresponding likelihood functions (the pdfs of the estimates conditioned on their origin [3]) are available.

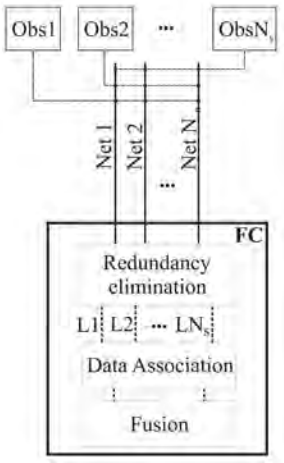


Fig. 2. Block schematic showing the hierarchical decomposition of the problem.

The remaining tracks after step 1 are reshuffled by common source s into lists according to the network they came on. For each s a search is done for "common origin" tracks across these lists using the LR association criterion of [3]. This criterion has to account for the dependence of the tracks (since common origin tracks have common measurements) according to (3). The search for these redundant tracks is done with MDA on (up to) N_n lists for each s. Tracks associated correspond to duplications of the same source reports sent via different networks. The best within each associated set (which is the most recent one) is kept. This step eliminates[4] the duplications in the set of tracks at the FC. Special attention should be given to incomplete associations across the N_n lists because event reports might not be sent via all the networks.

2. Track association across source lists.

At this point each source list contains the latest estimate for each event available from that source. The MDA will associate the elements across the N_s source lists (with at most one from each list) using the LR criterion function from [3]. The errors across the list elements are uncorrelated because there is no process noise and they have no common measurements.

3. Fusion of common origin tracks.

The tracks from different sources that have been designated by the MDA as having a common origin (same event) are fused. This is done according to the ML criterion from [9].

This decomposition drastically reduces the dimensionality of the problem from $N_s \times N_n$ to N_s problems of dimension N_n and one of dimension N_s. If the observer's track ID was available at the FC (via "ideal" networks), one would have only the second stage (item 2 above). The hierarchical approach avoids the need for an unnecessarily large single problem (the one formed by considering all the lists formed using the network IDs) in the case of the real-world networks and reduces the problem to two subproblems of the size one would have with ideal networks.

[4]Strictly speaking, this is a statistical testing approach subject to a maximum allowable (small) probability of error—incorrectly keeping a redundant track—according to which the test threshold is selected. This test then maximizes the power of the test—the probability of eliminating truly redundant tracks. However, this power (probability of eliminating truly redundant tracks) depends on the actual separation between neighboring distinct tracks.

4. ASSOCIATION CRITERIA

The following criteria can be used for track-to-track association:

i) Normalized distance squared (NDS). This criterion will be shown to be significantly inferior to the LR criterion. The reason for this is that large covariances reduce the NDS without imposing any penalty in view of the large uncertainty. Also, the use of NDS for associating tracks from more than 2 lists leads to necessarily heuristic approaches (see [12, 13]).

ii) Likelihood functions (LF). Since LF are pdf—with a physical dimension—they cannot be used for comparing associations of different number of tracks; this approach will not be pursued in this paper.

iii) Likelihood ratios (LR). These are physically dimensionless quantities and, consequently, allow comparison of associations of different number of tracks.

4.1. The Likelihood Ratio for Association

4.1.1. Removal of redundant tracks from the same source that arrived through different networks

Before using the MDA for removal of redundant tracks from the same source that arrived through different networks, a gating test should be used to select the candidates. The test will be based on the normalized distance squared (e.g., [5]) between pairs of tracks. Denoting the tracks now with full—triple—indexing, their normalized distance should be below a threshold, i.e.,

$$D(\hat{x}_{t_i,n_i,s}, \hat{x}_{t_j,n_j,s}) \triangleq (\hat{x}_{t_i,n_i,s} - \hat{x}_{t_j,n_j,s})' [T_{(t_i,n_i,s),(t_j,n_j,s)}]^{-1} (\hat{x}_{t_i,n_i,s} - \hat{x}_{t_j,n_j,s}) < c \tag{4}$$

where

$$T_{(t_i,n_i,s),(t_j,n_j,s)} \triangleq \operatorname{cov}(\hat{x}_{t_i,n_i,s} - \hat{x}_{t_j,n_j,s}) = P_{t_i,n_i,s} + P_{t_j,n_j,s} - P_{(t_i,n_i,s),(t_j,n_j,s)} - P'_{(t_i,n_i,s),(t_j,n_j,s)} \tag{5}$$

and (following [4], Sec. 8.4) $P_{t_i,n_i,s}$ is the track covariance corresponding to track $\hat{x}_{t_i,n_i,s}$ and $P_{(t_i,n_i,s),(t_j,n_j,s)}$ is the cross-covariance of the tracks with the indicated indexes. The elements of the cross-covariance are obtained according to (3) in the present problem.

The common origin likelihood function for a set of r tracks, composed of q non-dummies and $r-q$ dummies, from source s that arrived on different networks

$$\mathcal{T}_i = \{(t_{i_1}, n_{i_1}, s), \ldots, (t_{i_q}, n_{i_q}, s)\} \tag{6}$$

is given, under a diffuse prior assumption, by [3]

$$\Lambda(\mathcal{H}_{(t_{i_1}, n_{i_1}, s), \ldots, (t_{i_q}, n_{i_q}, s)}) = \frac{1}{V}\mathcal{N}[\hat{\mathbf{x}}_{\mathcal{T}_i}; 0, \mathbf{P}_{\mathcal{T}_i}]\mu_{\text{ex}}^{r-q}(P_d)^q(1-P_d)^{r-q} \tag{7}$$

where V is the (large) volume of the state space (in which the true common state is assumed uniformly distributed), P_d is the detection probability of an event, μ_{ex} is the "spatial density of extraneous targets," as shown in [2] using a spatial Poisson distribution. This density can be taken as the expected number of tracks (true and false) divided by V, and

$$\hat{\mathbf{x}}_{\mathcal{T}_i} \triangleq \begin{bmatrix} \hat{x}_{t_{i_2}, n_{i_2}, s} - \hat{x}_{t_{i_1}, n_{i_1}, s} \\ \vdots \\ \hat{x}_{t_{i_q}, n_{i_q}, s} - \hat{x}_{t_{i_1}, n_{i_1}, s} \end{bmatrix} \tag{8}$$

is a stacked $(q-1)n_x$-vector (with n_x the dimension of x), whose covariance has diagonal blocks

$$(\mathbf{P}_{\mathcal{T}_i})_{j-1,j-1} = T_{(t_{i_1}, n_{i_1}, s), (t_{i_j}, n_{i_j}, s)}, \qquad j = 2, \ldots, q \tag{9}$$

where $T_{(t_{i_1}, n_{i_1}, s), (t_{i_j}, n_{i_j}, s)}$ is given by (5), and the off-diagonal blocks are given by

$$(\mathbf{P}_{\mathcal{T}_i})_{k-1,j-1} = P_{t_{i_1}, n_{i_1}, s} - P_{(t_{i_k}, n_{i_k}, s), (t_{i_1}, n_{i_1}, s)} - P_{(t_{i_j}, n_{i_j}, s), (t_{i_1}, n_{i_1}, s)} + P_{(t_{i_k}, n_{i_k}, s), (t_{i_j}, n_{i_j}, s)}, \quad k, j = 2, \ldots, q; \quad k \neq j. \tag{10}$$

Since the comparisons might have to be made between hypotheses containing different numbers of tracks, likelihood functions cannot be used since, being pdf based, they have different physical dimensions for different numbers of tracks [5] and thus cannot be compared. Consequently, likelihood ratios have to be used. The likelihood ratio is obtained by dividing the above likelihood function by the joint pdf of the r track estimates under consideration, under the hypothesis that they are not of common origin. Given r tracks, the first one can be assumed uniformly distributed in V and the rest, which should be in its neighborhood, are again assumed to have a pdf given by the "spatial density of the extraneous targets," μ_{ex} (this is a consequence of the analysis presented in [2] using a spatial Poisson process).

Thus the LR will be

$$\mathcal{L}(\mathcal{H}_{(t_{i_1}, n_{i_1}, s), \ldots, (t_{i_q}, n_{i_q}, s)}) = \frac{1}{\mu_{\text{ex}}^{q-1}}\mathcal{N}[\hat{\mathbf{x}}_{\mathcal{T}_i}; 0, \mathbf{P}_{\mathcal{T}_i}](P_d)^q(1-P_d)^{r-q} \tag{11}$$

Note that the use of non-unity P_d "penalizes" incomplete associations. This becomes necessary for the costs obtained for this problem, as full tracks may yield better costs when split. For example, suppose a 4-D set of tracks having common origin $\{i_1, i_2, i_3, i_4\}$ yields cost C_{complete}, and a partition of two feasible partial track associations, $\{i_1, i_2, 0, 0\}$ and $\{0, 0, i_3, i_4\}$, yields costs C_{split_1} and C_{split_2} satisfying $C_{\text{complete}} < C_{\text{split}_i}$ for $i = 1, 2$ but $C_{\text{complete}} > C_{\text{split}_1} + C_{\text{split}_2}$. Then the split tracks can minimize the cost, although providing a less accurate solution. This undesirable phenomenon has been observed a number of times and the use of P_d in the LR cost function will avoid it.

The cost function to be used by the assignment algorithm in associating across the N_n lists is the negative of the logarithm of (11). This covers both complete associations (of N_n-tuples) as well as incomplete associations of q-tuples ($q < N_n$). In the latter case the "dummy element" [18] (i.e., no track) is chosen from $N_n - q$ lists; for these elements the likelihood ratios are taken as unity and consequently, they do not modify (11). The cost calculation for an association containing q non-dummy elements requires inversion of a $q \cdot n_x \times q \cdot n_x$ matrix.

Note that an association should have a negative cost if the (non-dummy) tracks in it are more likely to have a common origin than not. For $q = 1$, i.e., when a single track is associated with dummies (hence it is unassociated), the cost for this should be larger than an association with negative cost. Consequently an "unassociation" will be given zero cost and also associations with positive costs will be discarded, implementing an implicit fine gating.

From the LR cost formulation above it can be seen that, in terms of the computational complexity required, Network MDA is a harder problem than sequential MDA/MHT. In the case of having S lists with m_S reports in each, and each report contains a vector of data of dimension n, the sequential MDA/MHT calculates the association cost adding one list at a time. Each element of the first list does pass the gating test with a certain proportion, α, of the reports in the following list, and for the cost calculation the inversion of a matrix of size n is required; hence the computation requirement is $m_S(\alpha m_S)n^3$ for the first two lists. When the third list is added the requirement is $m_S(\alpha m_S)^2 n^3$ as a consequence of the reports in the third list passing the gating test, what accounts for the exponential growth of the hypotheses. Finally when the last list is added, the number of operations required is $m_S(\alpha m_S)^{S-1}n^3$, so the number of operations required is $O(m_S^S \alpha^{S-1} n^3)$. On the other hand, for the networked MDA, the static nature of the problem makes the cost calculation require the inversion of a $(S-1)n$ matrix, which renders a complexity of $O(m_S^S \alpha^{S-1}((S-1)n)^3)$. The cost calculation complexity of the sequential m-best 2D algorithm is much lower, as each list incorporated does also give birth to αm_S^2 associations, out of which roughly m_S are kept after the

2D association. Thus the number of cost calculations involved is $O(m_S(\alpha m_S)(S-1))$.

4.1.2. Track association across source lists

The common origin likelihood ratio for a set of N_s tracks (one from each source list), composed of q non-dummies and $N_s - q$ dummies, from different sources s_{j_l}, $l = 1,\ldots,q$

$$\mathcal{T}_j = \{(t_{j_1}, n_{j_1}, s_{j_1}), \ldots, (t_{j_q}, n_{j_q}, s_{j_q})\} \tag{12}$$

is given, similarly to (11), by

$$\mathcal{L}(\mathcal{H}_{(t_{j_1},n_{j_1},s_{j_1}),\ldots,(t_{j_q},n_{j_q},s_{j_q})}) = \frac{1}{\mu_{\text{ex}}^{q-1}} \mathcal{N}[\hat{\mathbf{x}}_{\mathcal{T}_j}; 0, \mathbf{P}_{\mathcal{T}_j}](P_d)^q (1-P_d)^{N_s-q} \tag{13}$$

where $\hat{\mathbf{x}}_{\mathcal{T}_j}$ is a stacked $(q-1)n_x$-vector as in (8) except that here all the tracks are from different sources.

The covariance in (13) has diagonal blocks

$$(\mathbf{P}_{\mathcal{T}_j})_{k-1,k-1} = T_{(t_{j_1},n_{j_1},s_{j_1}),(t_{j_k},n_{j_k},s_{j_k})} = P_{t_{j_1},n_{j_1},s_{j_1}} + P_{t_{j_k},n_{j_k},s_{j_k}}, \quad k = 2,\ldots,q. \tag{14}$$

Differing from (5), there are no cross-covariance terms in (14) as these tracks are from different sources and there is no process noise. Consequently, the off-diagonal blocks are

$$(\mathbf{P}_{\mathcal{T}_j})_{k-1,l-1} = P_{t_{j_1},n_{j_1},s_{j_1}}, \qquad k,l = 2,\ldots,q; \quad k \neq l. \tag{15}$$

4.2. The Normalized Distance Criterion for Association

4.2.1. Removal of redundant tracks from the same source that arrived through different networks

In addition to using the LR criterion, the simpler NDS (normalized distance squared) [4], also known as "Chi-square," which is the (negative of the) exponent of the likelihood function, will be considered. The NDS between a pair of tracks coming from the same observer through two different networks is defined as

$$D(\hat{x}_{t_{i_j},n_{i_j},s}, \hat{x}_{t_{i_k},n_{i_k},s}) \triangleq (\hat{x}_{t_{i_j},n_{i_j},s} - \hat{x}_{t_{i_k},n_{i_k},s})' [T_{(t_{i_j},n_{i_j},s),(t_{i_k},n_{i_k},s)}]^{-1} (\hat{x}_{t_{i_j},n_{i_j},s} - \hat{x}_{t_{i_k},n_{i_k},s}) \tag{16}$$

which is the LHS of (4). The covariance matrix is defined as in (5) with nonzero cross-covariance matrices due to the common origin of the tracks.[5]

As before, a track to dummy association will be given zero cost. However, due to the positiveness of the distance, the use of zero cost for the association to dummies implies that the best (least cost) assignment will consist of only track to dummy associations with zero cost. To avoid this problem and to implement the gating (4) at the same time, the cost for each association pair is calculated as

$$C_{j,k} = D_{j,k} - \chi^2_{n_x}(\alpha) \tag{17}$$

where $\chi^2_{n_x}(\alpha)$ is a level α threshold (usually big enough, say $\alpha > .99$) obtained from χ^2 tables with n_x degrees of freedom, and $D_{j,k}$ corresponds to the distance between tracks j and k (with abbreviated notation) as specified in (16). Then, if the distance $D_{j,k}$ is greater than the threshold, an assignment with positive resulting cost will never be selected, since an assignment to a dummy offers better (lower) cost. In case when the distance is smaller than the threshold, smaller distances will yield more negative costs than bigger distances, making them more attractive for assignment.

Since this distance criterion, defined above for a single pair, will be shown to yield inferior performance compared to the LR criterion, we will not pursue it any further. The use of the distance criterion for more than two tracks was discussed in [12].

4.2.2. Track association across source lists

The cost of associating a pair of tracks from different lists is the same as defined in (16) (suitably modifying the list indices), with the covariance matrix defined as

$$T_{(t_{j_l},n_{j_l},s_{j_l}),(t_{j_k},n_{j_k},s_{j_k})} = P_{t_{j_l},n_{j_l},s_{j_l}} + P_{t_{j_k},n_{j_k},s_{j_k}} \tag{18}$$

which is similar to (14).

Similarly to the removal of redundant tracks case, the cost of each association pair is defined here by subtracting a suitably defined threshold as in (17).

5. MULTIDIMENSIONAL ASSIGNMENT ALGORITHMS

Once the reports are split into lists, for duplication elimination in the first stage and for selection of candidates for fusion in the second stage, a multidimensional assignment problem [18] needs to be solved. A comparison of three different methods is carried out, one based on Lagrangean Relaxation, another based on a sequential calculation of m-best 2D assignments and the last one based on Linear Programming Relaxation.

5.1 Multidimensional Assignment Problem

The Multidimensional Assignment problem, also known as the S-D Assignment problem, consists of partitioning $S \geq 3$ lists of reports into S-tuples of report associations (RA) in a way that every report of every list is used, and that it is used only once. This problem is known to be NP-hard, which motivates the use of suboptimal methods.

[5] Note that we are testing whether these tracks from the same source represent the same event.

To allow for missed detections and false alarms (hence lists of different size) each list also contains a "dummy" element on which the above constraints do not apply. The inclusion of these reports converts the problem into a Generalized S-D Assignment. For the generalized problem we can partition the set of candidate associations into two. One subset corresponds to real track report associations, that is, S-tuples containing at least two non-dummy reports. The second subset corresponds to associations, that contain only one non-dummy report, i.e., unassociated track reports.

Defining binary event variables $\rho_{i_1,\ldots,i_S}$ to take value 1 when $\mathcal{T}_i = \{(t_{i_1}, n_{i_1}, s_{i_1}), \ldots, (t_{i_S}, n_{i_S}, s_{i_S})\}$, the S-tuple of tracks, is associated and 0 otherwise, the problem can be recast as a Linear Binary Programming (LBP) problem

$$\min_{\{\rho\}} \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_S=0}^{n_S} c_{i_1 i_2 \ldots i_S} \rho_{i_1 i_2 \ldots i_S} \tag{19}$$

subject to

$$\begin{aligned}
&\sum_{i_2=0}^{n_2} \sum_{i_3=0}^{n_3} \cdots \sum_{i_S=0}^{n_S} \rho_{i_1 i_2 \ldots i_S} = 1, \qquad i_1 = 1,2,\ldots,n_1 \\
&\sum_{i_1=0}^{n_1} \sum_{i_3=0}^{n_3} \cdots \sum_{i_S=0}^{n_S} \rho_{i_1 i_2 \ldots i_S} = 1, \qquad i_2 = 1,2,\ldots,n_2 \\
&\qquad\qquad \vdots \\
&\sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_{S-1}=0}^{n_{S-1}} \rho_{i_1 i_2 \ldots i_S} = 1, \qquad i_S = 1,2,\ldots,n_S
\end{aligned} \tag{20}$$

where $c_{i_1 i_2 \ldots i_S}$ is the cost of each association, given by the negative log-likelihood ratio (NLLR)

$$c_{i_1 i_2 \ldots i_S} = -\ln \mathcal{L}(\mathcal{H}_{(t_{i_1}, n_{i_1}, s), \ldots, (t_{i_m}, n_{i_m}, s)}) \tag{21}$$

where this LR is defined in (11). The dummy element in each list has index 0.

The following sub-sections briefly describe the three algorithms used for solving the described MDA assignment.

5.2 Lagrangean Relaxation Based S-D Assignment

The LBP problem can be suboptimally solved by relaxing the constraints and using Lagrange multipliers for them, until a 2D problem is obtained. This reduced dimension association problem can be solved exactly by well known algorithms such as JVC, Auction, Relax, etc. Then the relaxed constraints can be added one by one, using again a 2D association algorithm. This approach has been extensively described [17, 18, 19]. Sketchily, the S-D assignment problem is solved as a series of relaxed 2D subproblems in two phases: 1) relaxation of constraints, and 2) update of the Lagrange multipliers and constraint enforcement. In the first phase, each constraint set $r = S, S-1, \ldots, 3$ is appended to the cost function using Lagrange multiplier $\mathbf{u}_r$. After relaxing constraint set 3 we have a 2D assignment problem, which in our case is optimally solved using the generalized auction algorithm [7] (up to a certain accuracy—the granularity of the auction). After this the constraint enforcement phase begins by computing a solution of the 3D problem consisting of the previous assignment and the third constraint set, using again a 2D generalized auction algorithm. Then the multipliers $\mathbf{u}_3$ are updated using a subgradient method. Similarly, the successive constraints $r = 4,5,\ldots,S$ are enforced via a 2D assignment algorithm, and the multipliers $\mathbf{u}_r$ are updated. These two steps are repeated until all the constraints are satisfied in the relaxed problem, in which case the solution is optimal, or until the feasible solution is of acceptable quality.

It has been found that most of the running time (usually more than 95% of it) spent in solving the association problem with this approach is consumed by the calculation of the costs $c_{i_1 i_2 \ldots i_S}$. To alleviate this, clustering strategies may be used [10]. In our case performing gating during the construction of the association tree prevents the exponential growth of non-matching track branches, avoiding the corresponding cost calculations.

5.3 Sequential m-best 2D Assignment

This method provides a heuristic approach to obtain a Generalized S-D assignment solution. It relies on the solution of a sequence of Generalized 2D assignment problems [18, 20], where a dummy report element is introduced in each list. These dummy reports, which allow for missed detections, are not subject to the constraints (as discussed above), and the cost of associating any report to them is defined to be zero (the cost of a "real association" is negative). As a result, this modification allows the association between lists with different numbers of reports, and also allows for poorly matching reports not to be associated among them but to dummy reports, i.e., stay unassociated.

The algorithm is started by associating two lists, using a 2D generalized assignment algorithm. In this step, not only the best solution is kept, but also the following $m-1$ best (in terms of the association cost) solutions are found using an adaptation of Murty's method [16, 20]. The goal is to find the m best solutions of this assignment problem. This is achieved by first finding the best solution, using the generalized 2D assignment algorithm, and then partitioning it into exhaustive non-overlapping subproblems of smaller dimension. These problems are solved by the previous assignment algorithm, and out of them the best one will correspond to the second best solution. Partitioning this problem again, and keeping the best solution of the list formed with all the previous partitions provides the desired best solutions of the problem.

An adaptation of Murty's algorithm is necessary to handle the use of dummy elements in the generalized 2D assignment problem. The main point of this adaptation

consists of keeping dummies after a dummy has been used as pivot in the partitioning process, as opposed to partitioning a solution when non-dummy reports are selected as pivots, in which case this non-dummy report is voided from some of the spawning assignments, and forced in the rest of them. The complexity of the algorithm can be reduced from $O(mn^4)$ to $O(mn^3)$ by a clever implementation of the partitioning process, the inheritance of variables from the assignment method, and by bounding the subproblem costs [15]. Further improvement can be obtained by switching the assignment solving algorithm as a function of the sparsity of the problem, and by parallelizing the algorithm [20].

The Sequential m-best 2D Assignment is started by associating 2 lists, usually lists 1 and 2, and obtaining the top m solutions. This initial problem is a plain Generalized 2D problem, where the cost of associating a pair of reports is calculated as discussed above. Each of the solutions of this problem consists of a set of 2-tuples:

$$T_2^{(r)} = \{\mathcal{T}_{2,1}, \ldots, \mathcal{T}_{2,q_{2,r}}\}, \qquad r = 1, \ldots, m \tag{22}$$

where $q_{2,r}$ is the number of associated pairs for solutions from the first 2 lists and

$$\mathcal{T}_{2,i} = \{(t_{i_1}, n_{i_1}, s_{i_1}), (t_{i_2}, n_{i_2}, s_{i_2})\}$$

is a 2D report association, the basic component of each solution.

To add a new list, another generalized 2D problem must be solved to match the new list elements with the associations in $T_2^{(r)}$, for each r. Each of the corresponding 2D assignment matrices will have dimension $q_{2,r} \times n_3$, where n_3 is the number of elements in list 3, and the cost for each element is calculated using the 3-tuple defined by

$$\mathcal{T}_{3,i} = \{\mathcal{T}_{2,j}, (t_{i_3}, n_{i_3}, s_{i_3})\} \tag{23}$$

where $j = 1, \ldots, q_{2,r}$ and $i = 1, \ldots, n_3$, with the cost function defined as before.

Rather than calculating m new solutions for each of the previous m best solutions, which would yield m^2 solutions from which to pick the top m, we initialize the list of problems with the m previous assignments. This makes the algorithm run only once, decomposing the solution which has the best cost at each time. Then the same m best associations will be obtained for considerably less computation [11].

After this second step, the obtained m best solutions are represented by

$$T_3^{(r)} = \{\mathcal{T}_{3,1}, \ldots, \mathcal{T}_{3,q_{3,r}}\}, \qquad r = 1, \ldots, m \tag{24}$$

where $q_{3,r}$ is the number of associated triplets for each solution, and

$$\mathcal{T}_{3,i} = \{(t_{i_1}, n_{i_1}, s_{i_1}), (t_{i_2}, n_{i_2}, s_{i_2}), (t_{i_3}, n_{i_3}, s_{i_3})\}$$

is a 3D report association.

The rest of the lists are incorporated using the same procedure. That is, for each of the m best solutions obtained after adding list k, a 2D association matrix with the costs of associating its RAs and the reports from list $k+1$ is calculated. Costs are again calculated using all the combinations of k-D RAs and list $k+1$ reports, together with (13) if there are more that two non-dummies, and setting 0 cost in case of having only one non-dummy element in the $(k+1)$-D RA. After all cost matrices corresponding to the previous best solutions and the new list are obtained, extended solutions formed by $(k+1)$-D RAs are calculated using the m-best algorithm with the mentioned cost matrices. After solving these problems, the m best solutions are found and the procedure is repeated with list $k+2$, and so on until the last list is incorporated. When the last list is finally incorporated, only the top solution is used as the resulting S-D association matrix, thus providing a "hard" solution to the association problem. Soft solutions that combine the m final associations may provide a better quality solution. This is currently under investigation.

In the results section, we will show that even for large values of m the quality of the solution for the problem considered does not improve from just taking the best solution at each step.

5.4 Linear Programming Based Assignment

The Linear Binary Programming problem defined by (19) and (20) can be relaxed to a Linear Programming (LP) problem by allowing non-integer values for the event variables ρ. This relaxed problem can be solved using several efficient LP algorithms, but the integrality of the solution is not ensured. This brings up the question of what to do with the fractional assignments. In general for the assignment problem, the occurrence of these fractional solutions (assignments) is rare. Thus for the present work we consider the fused fractional assignments as fractional tracks, i.e., after fusion an assignment with $0 < \rho_{i_1 i_2 \ldots i_S} < 1$ will count as (a fractional) $\rho_{i_1 i_2 \ldots i_S}$ track.

The number of variables involved in the LP problem is proportional to the product of the number of reports in each list. For example, for a problem with 4 lists and 10 tracks per list, the number of variables is greater than 10^4. To reduce the number of variables, we consider only those variables with zero or negative cost (i.e., we perform an implicit gating). The resulting reduced set of indexes is

$$\Xi = \{(i_1 i_2 \ldots i_S) : c_{i_1 i_2 \ldots i_S} \leq 0,\ i_1 = 0, \ldots, n_1,\ \ldots,\ i_S = 0, \ldots, n_S\}. \tag{25}$$

Then the reduced LP problem becomes

$$\min_{\{\rho\}} \sum_{(i_1 i_2 \ldots i_S) \in \Xi} c_{i_1 i_2 \ldots i_S} \rho_{i_1 i_2 \ldots i_S} \tag{26}$$

subject to

$$\begin{aligned}
&\sum_{\{(i_2 i_3 \ldots i_S):(i_1 i_2 \ldots i_S)\in\Xi\}} \rho_{i_1 i_2 \ldots i_S} = 1, \qquad i_1 = 1,2,\ldots,n_1 \\
&\sum_{\{(i_1 i_3 \ldots i_S):(i_1 i_2 \ldots i_S)\in\Xi\}} \rho_{i_1 i_2 \ldots i_S} = 1, \qquad i_2 = 1,2,\ldots,n_2 \\
&\qquad\qquad \vdots \\
&\sum_{\{(i_1 i_2 \ldots i_{S-1}):(i_1 i_2 \ldots i_S)\in\Xi\}} \rho_{i_1 i_2 \ldots i_S} = 1, \qquad i_S = 1,2,\ldots,n_S.
\end{aligned} \tag{27}$$

The solver used for this work is LPSolve [6], a free GNU program which implements a primal-dual method. This solver will be shown to provide fast solutions when the number of variables involved is below 10^4, although it is not of the self-dual family, which is reported to operate very efficiently for the 3D assignment problem [14].

6. SIMULATION RESULTS

The scenario consists of a set of events (missile launches) for which 2-D position, launch time and heading need to be estimated, using multiple sensors. The surveillance region covered by each sensors is $x \in [0,10000]$, $y \in [0,10000]$ for position, $\psi \in [0,30°]$ for heading. The launches occur randomly in the surveillance region, during a time interval $t \in [0,10]$. Each sensor receives measurements from each target on average every 10 units of time and transmits reports on average every 20 units of time. The time span of the scenarios is 200 units of time, and both redundancy elimination and fusion are performed every 20 units of time. The false report rate per sensor is P_f per unit time, so on the average there are $200P_f$ false track/event reports per sensor. Also, each of the communication networks has a probability P_r of delivering the report to the FC (its reliability). The measurements from each target received by each sensor are corrupted with white Gaussian noise, with standard deviations $\sigma_x = \sigma_y = 2500$, $\sigma_\psi = 10$, $\sigma_t = 3$ (uncorrelated components).

Scenario 1. The parameters that specify this scenario are: 2 events (launches), $N_n = 2$ networks with reliability $P_r = .5$, 2 sensors and false report rate (per unit time) $P_f = .01$. The differences in the values of interest between the two true event locations, headings and launch times, are $\Delta x = \Delta y = 1000$, $\Delta h = 6$, $\Delta t = 2$, respectively.

Figures 3 and 4 show the result of the use of the LR criterion and NDS criterion on this scenario, for 1000 Monte Carlo runs. They show the percentage of incorrect fusions (i.e., the fusion of two tracks with different origin) and incorrect eliminations (e.g., eliminating a report from a pair with different origins). It can be seen that when the LR criterion is used, as more data are obtained, the percentage of errors is reduced, despite the presence of false reports, which are handled correctly.

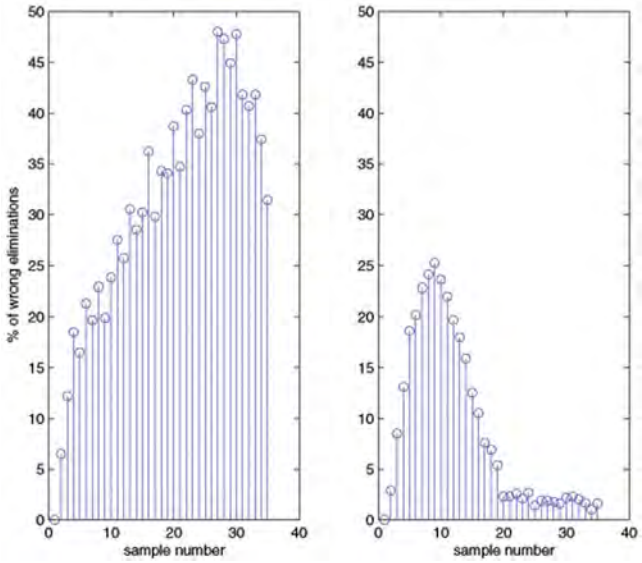


Fig. 3. Percentage of errors in the duplication elimination step using the NDS distance cost criterion (left) and LR cost criterion (right).

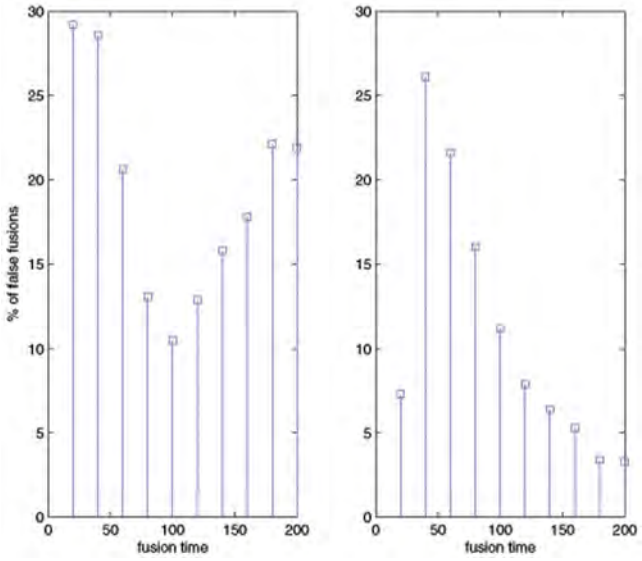


Fig. 4. Percentage of errors in the fusion candidate selection step using the NDS distance (left) cost criterion and LR cost criterion (right).

On the other hand, when the NDS criterion is used, the presence of false reports does affect both stages in a way that the errors committed reach very high levels. This is a consequence of the false reports, which have large standard deviations compared to the normal tracks and yield a low NDS with respect to almost any track. In the LR the presence of the standard deviation terms in the denominator compensates for this, and results in substantially fewer association errors.

Scenario 2. To characterize the behavior of the assignment methods presented in Section 5, different size problems are used, ranging from mid size problems (4D), to bigger problems, up to 7D. The parameters that govern this scenario are: 40 events, $N_n = 4$ networks,

and $N_s = 4,5,6,7$ sensors. The impact of using P_d is investigated, as well as the selection of $\mu_{ex} = n_{ex}/V$ (expressed as only its numerator, n_{ex}), which will be varied from the theoretical value in [2] (expected number of total tracks per unit volume) to just the number of false tracks divided by the surveillance region volume. For each of these problems 50 Monte Carlo runs were performed for each of the proposed search methods.

Due to the initial variance of the estimates of each observer and the surveillance region volume, this scenario is dense in the sense that several tracks gate with each other, making the association problem hard to solve. A coarse (very optimistic) approximation to the number of possible non-overlapping events is to divide the surveillance region volume by the product of n standard deviations of the estimates. At the initial time this is $10000 \times 10000 \times 30 \times 10/(n \times 2500) \times (n \times 2500) \times (n \times 10) \times (n \times 3)$ which gives approximately 2 for $n = 3$ and 10 for $n = 2$. As new measurements come in, the track estimates reduce their variance, and the number of feasible associations reduce, then, considering that at the final time the s.d. is approximately reduced by a factor of $\sqrt{5}$, it is possible to have up to 50 non-overlapping tracks for $n = 3$. Simulations show that each track from a list gates on average 8 tracks from any other list when the track estimates are of poor quality (large variance, early in the game) and about 5 tracks when the uncertainty of the tracks is reduced. That is, around 15–20% of the tracks are gated, so this can be used as a rough estimate of the sparsity.

In general, for any number of lists, the method that finds the lower costs is the Linear Programming based S-D assignment, which does rarely come up with a fractional solution: however this is at a computation time expense that grows with the number of negative cost associations that are fed to the LP solver (this number increases with the number of lists, the number of elements per list and the value of n_{ex}). The solution cost found by the LP solver is almost always lower than the true cost, due to the noise that causes some associations to give better matches than the truth. The Sequential m-best 2D does also usually find lower costs than the truth, but this behavior is dependent on the parameter values. For high n_{ex} this happens both for the cases of using and not using the P_d term, but for low n_{ex} the usage of P_d makes the algorithm find costs lower than the truth, while not using it renders costs higher than the truth. This effect is reduced as the final time is approached. This behavior is a consequence of the myopicity of the algorithm, as will be explained later. The Lagrangean Relaxation based approach usually finds solutions with costs higher that the two previous methods, which are still lower than the truth, but with the advantage of getting a higher number of full associations, an effect that will be further discussed in the sequel.

To quantify the quality of the fused tracks, a partition into 3 fusion categories is done. The first category represents the fusion of reports having identical origin and using one report from each possible list (observer), i.e., completely correct (CC) fusions. The second category represents fusions which have at least 2 reports with common origin, and the rest may be from different origins (but not very distant, due to gating) or not detected, i.e., partially correct (PC) fusions. The last category consists of fusions without any pair of reports coming from the same origin, i.e., completely incorrect (CI) fusions. The sum of the PC and CC fusions is taken as the reference to quantify the performance of the fusion. For this scenario, a number of PC+CC fusions of 40 is optimal, together with a value of 0 for the CI fusions. The number of PC+CC fusions will be used hereafter to measure the quality of the solutions obtained. A more thorough measure would involve also the CI fusions, but these values tend to zero for all the methods presented, and are not presented here for brevity.

Figures 5–7 show PC+CC fusion results for the 7 observers scenario using each of the proposed MDA algorithms. In all cases the solution at earlier times improves as n_{ex} is reduced, and the effect of using P_d is noticeable. For $n_{ex} = 1$ all the algorithms provide good quality solutions, while for higher values the sequential m-best 2D finds lower quality solutions. The usage of the P_d term does alleviate this, although for higher values of n_{ex} this is not enough to get good solutions out of this algorithm. In general the increase in the number of PC+CC fusions comes from a phenomenon to be called "track splitting," in which a CC association is divided into two or more PC associations, which provide an overall lower cost. On the other hand, a decrease in the number of PC+CC fusions is caused for large n_{ex} by the unattractiveness of the track to track association, so the majority of the tracks are associated with dummies, resulting in a smaller number of fusions.

The Sequential m-best 2D (Sm2D) algorithm was used with two different values of m, 1 and 10, with practically the same results, and contrary to expectations, increasing m does not necessarily improve the cost. This happens mainly due to three reasons:

i) The algorithm associates one list at a time, and the m-best solutions for a problem size like the one considered here (at least 40×40) are very similar (usually differ in only one report). So in general all the m-best associations surviving one of these steps are minor variations of a single association. This can be mitigated by using a much larger m, which in general will be a function of the problem size; however, the increase in the problem size implies a need for huge values that usually render the algorithm time-infeasible, so values of m bigger than 10 are not used.

ii) The hierarchical approach and the suboptimality of the calculations due to the fact that the data to be fused has already undergone association, which has some probability of producing errors, and can contain duplicate tracks. These redundant tracks affect the fu-

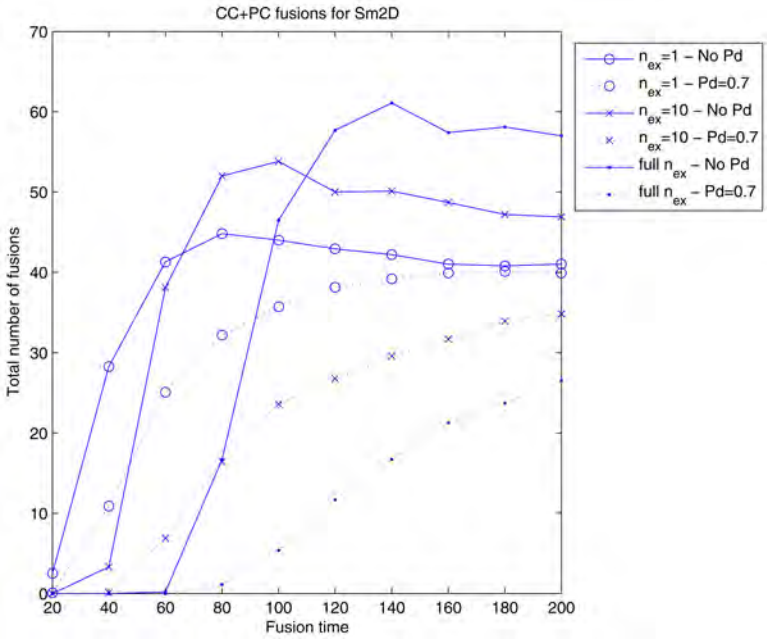


Fig. 5. Fusion quality for scenario 2, 7 observers, using S*m*2D.

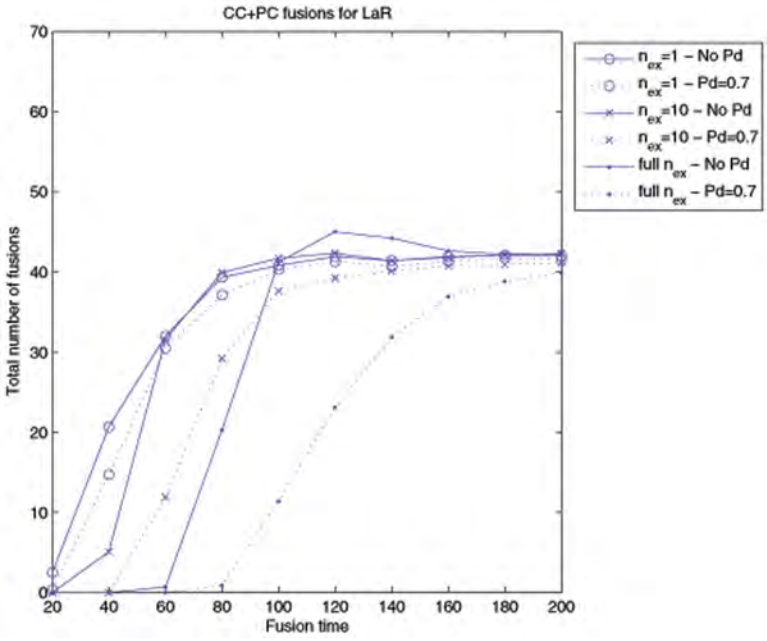


Fig. 6. Fusion quality for scenario 2, 7 observers, using LaR.

sion, and in general, the inclusion of redundant or noisy reports allows the cost to be lower than the true one. For example, if a track is observed by all 4 sensors, it is possible that the S*m*2D algorithm groups them into two different sets of 2, due to its "myopic (greedy) approach" for the sake of a lower cost. To alleviate this problem, the use of a "penalization" coefficient, P_d, was introduced in the cost definition, and proves to ameliorate the quality of the resulting tracks.

iii) The myopicity, which seems to be the main reason that, for reasonable values of m, the S*m*2D solution has a lower cost but more errors compared to the LaR. Specifically, the S*m*2D algorithm will not retain a positive cost that later could become negative (because it prefers "myopically" the zero or negative costs which fill the top m solutions). Another reason for the differences between S*m*2D and LaR *S*-D lies in the nature of the latter. There, in the relaxed cost calculations, a minimization operation is performed which usually rules out partial associations in favor of a full association. This full association is more likely to have better cost than the individual partials but not than the sum of the two complementary (split) partial associations that use the same tracks as the full association (see the example

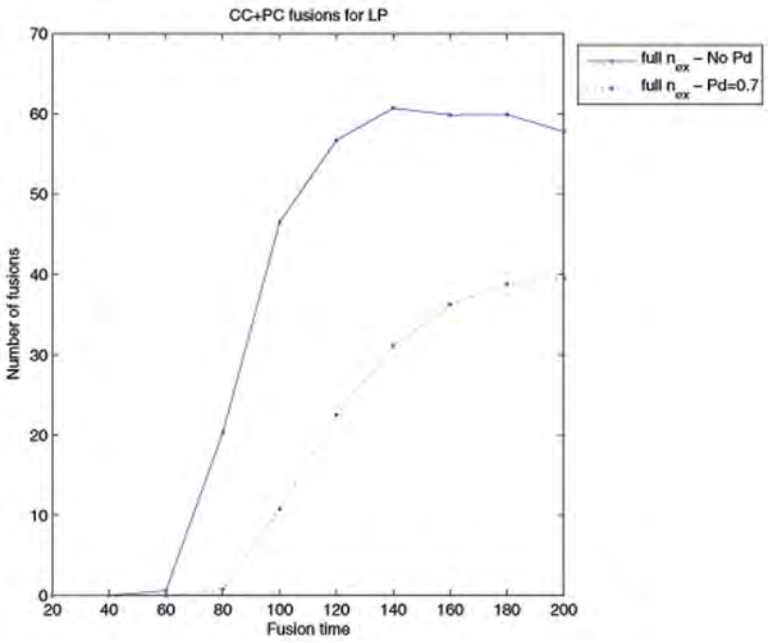


Fig. 7. Fusion quality for scenario 2, 7 observers, using LP (for low n_{ex} run-time is prohibitive—too many costs).

in Section 4.1.1). This feature naturally favors full associations to partial (split) ones, although the latter may yield lower costs when not penalized.

As previously stated, the inclusion of the P_d term in the cost function helps to alleviate the usual problem of having better costs for split tracks than for complete tracks (see Fig. 7), but it may overpenalize the incomplete associations, discarding some real associations (see Fig. 5). Continuing with the example in Section 4.1.1, and using costs obtained from a problem with $n_{\text{ex}} = 1$ and without using P_d, we have the following costs for a particular set of 4 tracks $\{i_1, i_2, i_3, i_4\}$ with common origin: $C_{\text{complete}} = C_{i_1,i_2,i_3,i_4} = -1.46$, while for the partition into two feasible partial associations one has $C_{\text{split}_1} = C_{i_1,i_2,0,0} = -.97$ and $C_{\text{split}_2} = C_{0,0,i_3,i_4} = -.83$. As before, we have $C_{\text{complete}} < C_{\text{split}_i}$ for $i = 1, 2$ but $C_{\text{complete}} > C_{\text{split}_1} + C_{\text{split}_2}$. Then the split associations can minimize the cost, although providing a less accurate solution. The Sm2D algorithm and the LP based method select this (cheaper but undesirable) track split, while the LaR algorithm does not. The reason for the LaR not selecting track splits lies in its suboptimality: at each constraint relaxation step a minimization is performed, fixing one track. In our example, the competition between the complete track and one of the partial tracks will happen when relaxing the constraints corresponding to i_3. At this point we will have costs $D_{i_1,i_2,i_3} = \min_{i_4}(C_{i_1,i_2,i_3,i_4} - \mu_{4,i_4})$, where μ_{4,i_4} is a Lagrange multiplier. It is very likely that both C_{i_1,i_2,i_3,i_4} and $C_{i_1,i_2,0,0}$ will survive and generate D_{i_1,i_2,i_3} and $D_{i_1,i_2,0}$. A further constraint relaxation will yield 2-D costs $B_{i_1,i_2,i_3} = \min_{i_3}(D_{i_1,i_2,i_3} - \mu_{3,i_3})$. Here the two associations will be competing one to one via the modified costs B, and it is usually the case that the Lagrange multipliers do not revert the cost majority relation between full and split associations, hence the association split_1 is discarded and the complete association is kept. As this association is feasible, and yields a better cost than split_2, the multiplier's update will penalize the usage of this split association, and hence causing the complete association to be selected after several iterations of the algorithm. This feature of the algorithm is undesirable in terms of getting the lowest possible cost, but in our case turns out to be an advantage, as it implicity forces a track completeness constraint. In other applications, where there are no full associations with common origin, this feature may yield degraded results.

REMARK The above discussion raises the issue of cost and optimization algorithm selection. The cost is really a surrogate for the "association accuracy" desideratum. However, no cost is an exact reflection of our desideratum and this motivates our detailed investigation of cost parameters and optimization algorithms.

Besides the quality of the solution, another fundamental aspect of these algorithms is run time, as suboptimality of the solution can be traded off for a speedup in the problem solution (otherwise a complete enumeration of the possibilities would find the optimal solution, at the expense of a huge run time). The algorithms used were coded in C++ and run on a P4-2.8 GHz computer.

Figures 8–10 correspond to run times for problems with 4 observers, 7 observers, and the ratio of computation time of the LaR based method and LP based method, over the time taken by the Sm2D method. Run times are split into two parts, time spent in cost calculation, and time spent purely in the optimization algorithm. As previously mentioned, all the algorithms have the best performance when $n_{\text{ex}} = 1$ and P_d is present,

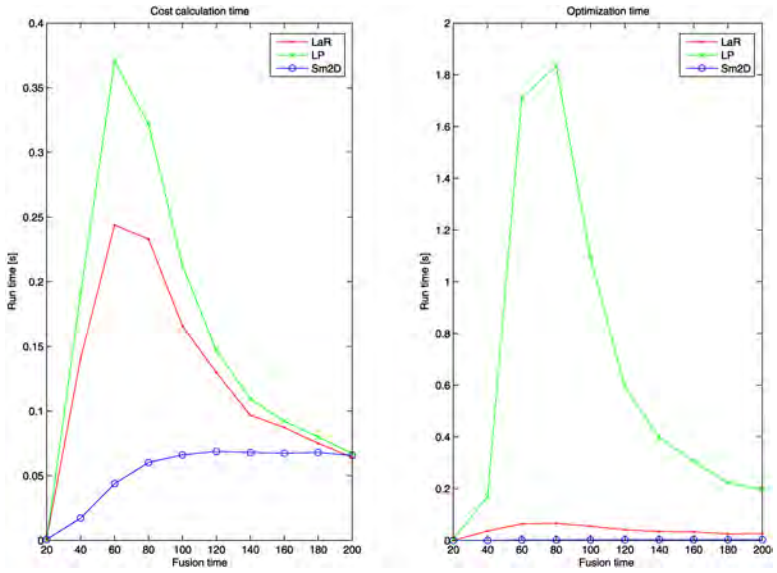


Fig. 8. Run time for a scenario with 4 observers, 4 networks and 40 launches, using $P_d = 0.7$ and $n_{\text{ex}} = 1$. The left figure shows the time spent in calculating the association costs, and the right figure shows the time spent in solving the minimization problem.

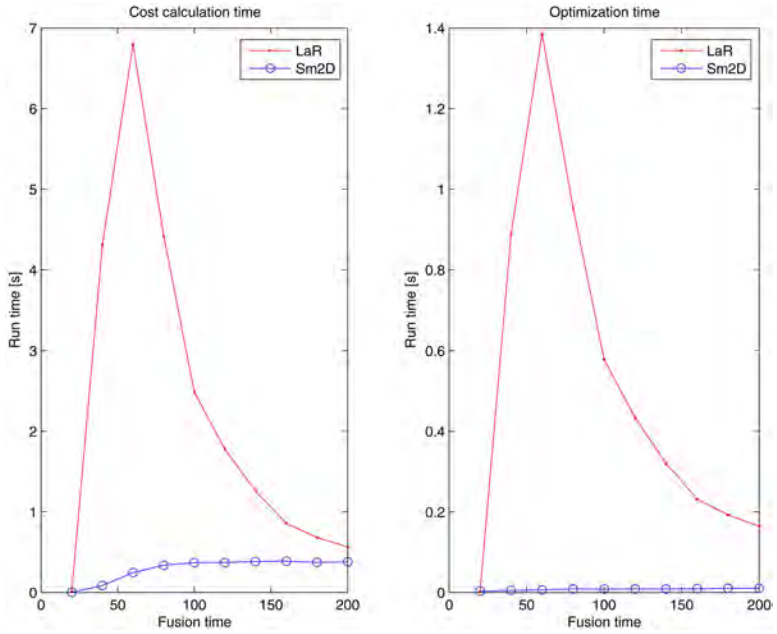


Fig. 9. Run time for a scenario with 7 observers, 4 networks and 40 launches, using $P_d = 0.7$ and $n_{\text{ex}} = 1$. Linear Programming time is not shown as it exceeds the times shown by at least 2 orders of magnitude—a consequence of the large number of negative cost associations for the scenario.

thus the presented results correspond to these parameter values.

The run time of the S*m*2D is the best, and its advantage over LaR and LP improves when the number of lists is increased. Also it does not change noticeably with the variation of the parameters P_d and n_{ex}, while for the other two methods the n_{ex} parameter affects the run time in diverse ways. For the LaR based method a smaller value of n_{ex} does increase the number of nonredundant tracks found after redundancy elimination, especially during the initial fusion times. Hence the feasible cost tree construction takes longer, as many reports gate with each other, making the time increase be polynomial. The time taken purely by the LaR minimization algorithm shows a linear increase with the tree size, so there is also a time increase in the minimization algorithm, but the cost calculation time clearly dominates the overall run time. For the LP based method the cost calculation follows the same pattern, but the minimization part does not show a linear increase as n_{ex} is decreased, as this has not only the effect of increasing the tree size, but also the number of negative association costs. For problems with more than $5 \cdot 10^4$ negative cost associations, the LP algorithm takes too long to run, and

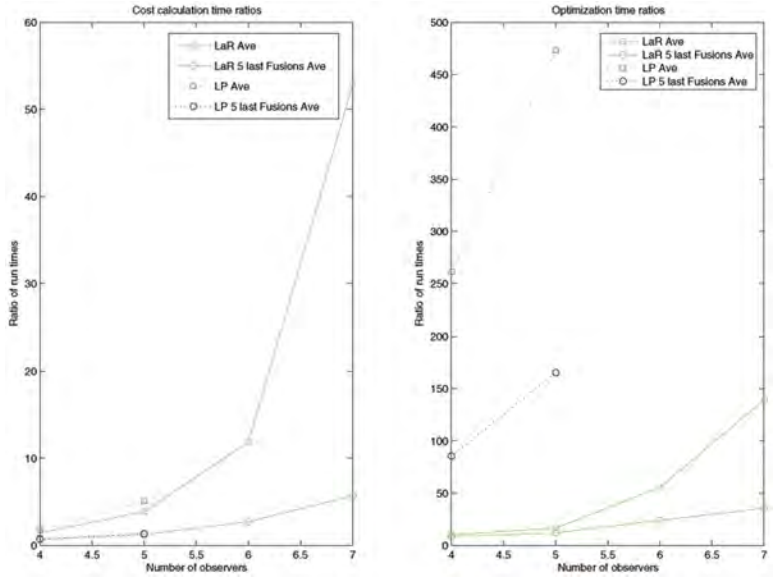


Fig. 10. Ratio of run times for a scenario with 7 observers, 4 networks and 40 launches, using $P_d = 0.7$ and $n_{ex} = 1$. The reference run time corresponds to the Sequential m-best 2D algorithm. There are 2 different measures of time, the average of the time ratios over the 10 fusion times, and the average of the time ratio over the last 5 fusion times, as from the previous two figures it can be seen that most of the time is spent in the first 5 fusion times, inflating the average.

dominates over the cost calculation time. This is not an uncommon situation for the case of having more than 4 lists, 40 elements per list and n_{ex} values below 10. In Figure 10 the run time ratios for the latter method is shown only for the case of having 4 and 5 lists, as for higher values the time taken by the LP algorithm is so large that its use is precluded for more than 5 observers.

7. SUMMARY AND CONCLUSIONS

This work compared three MDA algorithms when applied to a benchmark problem. A hierarchical scheme was developed to eliminate the duplicate reports from the same observer transmitted through N_n different networks, and then to select which reports from each of the N_s observers will be fused. This approach drastically reduces the dimensionality of the problem from $N_s \times N_n$ to N_s problems of dimension N_n and one of dimension N_s. This dimensionality reduction allows the use of algorithms like Lagrangean Relaxation and Linear Programming, which are infeasible for high density and high dimension problems due to run time limitations. The results for the comparison of different association criteria indicate a significant performance improvement when the Likelihood Ratio criterion is used vs. the NDS. Due to this, and the intrinsic 2D nature of the NDS cost criterion, this method is not used for the higher dimension problems presented. The Sequential m-best 2D assignment algorithm was found to be the fastest. However, its performance does not improve as the number m of best solutions kept at each stage is increased, and the cost does not necessarily improve with this increase in the number of best solutions kept. This behavior is mainly due to the "myopicity" of the approach, which also showed a tendency for preferring incomplete associations. The usage of a penalization term, P_d, for the cost calculation alleviates this drawback, by discouraging incomplete associations. The Linear Programming approach provides the lowest cost solutions, but they are not necessarily the best quality solutions (in terms of association accuracy), at a time expense similar to the Lagrangean Relaxation approach for cost calculation, but much higher for the optimization algorithm when the number of negative association costs is large. The LP method also has the drawback of giving non-integer solutions from time to time (less than 5% of the time a non-integer solution was observed). Both the LP and Sequential m-best 2D algorithms yield solutions that depend highly on the value of the number of extraneous targets (a parameter in the LR cost) n_{ex} used, giving better results for small n_{ex}, close to the number of false tracks. On the other hand, the Lagrangean Relaxation approach has proven to be the most robust method that works consistently for almost all parameter values.

Since, in general, the cost function parameters n_{ex} and P_d are not exactly known, the Lagrangean Relaxation algorithm is proposed as the algorithm to use as its robustness pays off for the run time increase when compared to the Sequential m-best 2D algorithm, and it exceeds the performance of the Linear Programming algorithm both in run time and solution quality when the number of lists grows above 4.

APPENDIX

A. Gating implementation

Given S lists with m_S event reports in each, we form an association tree which will contain all the

possible S-tuples. During the construction of the tree, a coarse gating strategy is used to discard unfeasible associations, and reduce the size of the tree. The tree constructed in this implementation is of depth S, with each level corresponding to a list.

The construction of the tree is recursive. Beginning at list 1, report 1 $(1,1)$, we continue with $(2,1)$ and so on until $(S,1)$. Each time a report is incorporated we test if it falls inside the gates defined by the previous reports in the tuple. Next we exhaust list S reports, fixing the $S-1$ previous reports, and generate $m_S - 1$ branches by selecting (S,i) for $i = 2,\ldots,m_S$. In the same way, fixing the first $S-2$ reports we exhaust list $S-1$, and at each value of $(S-1,j)$ all the values of (S,i) are also exhausted. We continue with the process until all the lists are incorporated.

There are two possible gating methods, one based on the NDS of Section 4.2.1, and other based on the individual coordinates, where the distance between scalar components of the report is compared to a threshold proportional to its standard deviation. In this way, if a report does not pass the gating test all the branches fanning from this node are discarded.

B. Inversion of the stacked covariance matrix

In equations (11) and (13), the LR cost calculation for association requires the inversion of the correlation matrix $\mathbf{P}_{\mathcal{T}_i}$ and the determinant of this inverse. The dimension of this matrix grows linearly with the number of tracks, so efficient ways of calculating this inverse are of great interest.

Using a permutation matrix as in [2] the vector of differences $\hat{\mathbf{x}}_{\mathcal{T}_i}$ can be transformed to a vector where the differences are among consecutive tracks

$$\hat{\mathbf{x}}_{\mathcal{T}_i} \triangleq \begin{bmatrix} \hat{x}_{t_{i_2},n_{i_2},s_{i_2}} - \hat{x}_{t_{i_1},n_{i_1},s_{i_1}} \\ \hat{x}_{t_{i_3},n_{i_3},s_{i_3}} - \hat{x}_{t_{i_2},n_{i_2},s_{i_2}} \\ \vdots \\ \hat{x}_{t_{i_q},n_{i_q},s_{i_q}} - \hat{x}_{t_{i_{q-1}},n_{i_{q-1}},s_{i_{q-1}}} \\ \hat{x}_{t_{i_1},n_{i_1},s_{i_1}} - \hat{x}_{t_{i_q},n_{i_q},s_{i_q}} \end{bmatrix}. \tag{28}$$

For the particular case of data association prior to fusion, the reports are uncorrelated (see Section 4.1.2), so the corresponding correlation matrix will be block tridiagonal, with block size n_x equal to the dimension of x_{t_i,n_i,s_i}. Following [8], the correlation matrix corresponding to q tracks can be written as

$$\mathbf{P}_{\mathcal{T}_i} = \begin{bmatrix} R_{11} & R_{12} & & & \\ R_{21} & R_{22} & R_{23} & & \\ & R_{32} & \ddots & \ddots & \\ & & \ddots & R_{q-2q-2} & R_{q-2q-1} \\ & & & R_{q-1q-2} & R_{q-1q-1} \end{bmatrix}$$

$$= \begin{bmatrix} D_1 & * & & & \\ E_1 & D_2 & * & & \\ & E_2 & \ddots & \ddots & \\ & & \ddots & D_{q-2} & * \\ & & & E_{q-2} & D_{q-1} \end{bmatrix} \tag{29}$$

and decomposed into block Cholesky factors

$$\mathbf{L}_{\mathcal{T}_i} = \begin{bmatrix} \tilde{D}_1 & & & & \\ \tilde{E}_1 & \tilde{D}_2 & & & \\ & \tilde{E}_2 & \ddots & & \\ & & \ddots & \tilde{D}_{q-2} & \\ & & & \tilde{E}_{q-2} & \tilde{D}_{q-1} \end{bmatrix} \tag{30}$$

where $\tilde{D}_i$, $i = 1,\ldots,q-1$ are lower triangular and

$$\begin{aligned} D_1 &= \tilde{D}_1\tilde{D}_1^T \\ E_i &= \tilde{E}_i\tilde{D}_i^{-T}, \qquad i = 1,\ldots,q-2 \\ D_i - \tilde{E}_i\tilde{E}_i^T &= \tilde{D}_i\tilde{D}_i^T, \qquad i = 2,\ldots,q-1. \end{aligned} \tag{31}$$

Then, an algorithm for overwriting the block tridiagonal matrix blocks with the corresponding block Cholesky factors is given by

$$\begin{aligned} &\textbf{for} \quad i = 1,\ldots,q-2 \\ &\qquad D_i \leftarrow \hat{D}_i = \mathrm{Chol}(D_i); \qquad n_x^3/3 \text{ ops.} \\ &\qquad E_i \leftarrow \hat{E}_i = E_i D_i^{-T}; \qquad n_x^3 \text{ ops.} \\ &\qquad D_{i+1} \leftarrow D_{i+1} - E_i E_i^T; \qquad n_x^3 \text{ ops.} \\ &\textbf{end} \\ &D_{q-1} \quad \leftarrow \hat{D}_{q-1} = \mathrm{Chol}(D_{q-1}); \qquad n_x^3/3 \text{ ops.} \end{aligned} \tag{32}$$

Thus the total operation count of the algorithm is

$$\tfrac{1}{3}(q-1)n_x^3 + 2*(q-2)n_x^3 \approx \tfrac{7}{3}(q-1)n_x^3 \tag{33}$$

that is, the required number of operations for obtaining the Cholesky decomposition is linear in the number of blocks. The inverse of $\mathbf{P}_{\mathcal{T}_i}$ is not explicitly required, as it is used in a quadratic form problem

$$\begin{aligned} \hat{\mathbf{x}}_{\mathcal{T}_i}^T \mathbf{P}_{\mathcal{T}_i}^{-1} \hat{\mathbf{x}}_{\mathcal{T}_i} &= \hat{\mathbf{x}}_{\mathcal{T}_i}^T (\mathbf{L}_{\mathcal{T}_i}\mathbf{L}_{\mathcal{T}_i}^T)^{-1} \hat{\mathbf{x}}_{\mathcal{T}_i} \\ &= (\mathbf{L}_{\mathcal{T}_i}^{-1}\hat{\mathbf{x}}_{\mathcal{T}_i})^T (\mathbf{L}_{\mathcal{T}_i}^{-1}\hat{\mathbf{x}}_{\mathcal{T}_i}) \\ &= \mathbf{y}_{\mathcal{T}_i}^T \mathbf{y}_{\mathcal{T}_i} \end{aligned} \tag{34}$$

so back-substitution can be used to solve for $\mathbf{y}_{\mathcal{T}_i}$ in

$$\mathbf{L}_{\mathcal{T}_i}\mathbf{y}_{\mathcal{T}_i} = \hat{\mathbf{x}}_{\mathcal{T}_i} \tag{35}$$

requiring approximately $n_x(q-1)$ operations.

The determinant of $\mathbf{P}_{\mathcal{T}_i}^{-1}$ can be calculated as one over the determinant of $\mathbf{P}_{\mathcal{T}_i}$, and this is obtained from the diagonal of $\mathbf{L}_{\mathcal{T}_i}$ in just $(q-1)$ operations.

Overall, for the LR cost calculation for the selection of tracks to fuse the dominating term in the operation count is $7/3(q-1)n_x^3$, which, as previously said, is linear in the number of blocks.

For the case of the redundancy elimination, the correlation matrix is full, and thus the operation count corresponds to the Cholesky decomposition of the matrix, $((q-1)n_x)^3/3$, which is no longer linear in the number of blocks.

REFERENCES

[1] J. Areta, Y. Bar-Shalom and M. Levedahl
A hierarchical benchmark association problem in missile defense.
In *Proceedings SPIE Conference on Tracking Small Targets*, vol. 5913–38, San Diego, CA, Aug. 2005.

[2] Y. Bar-Shalom, S. S. Blackman and R. J. Fitzgerald
The dimensionless score function for measurement to track association.
IEEE Transactions on Aerospace and Electronic Systems, **43**, 1 (Jan. 2007), 392–400.

[3] Y. Bar-Shalom and H. Chen
Multisensor track-to-track association for tracks with dependent errors.
Journal of Advances in Information Fusion, **1**, 1 (July 2006), 3–17.

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

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

[6] M. Berkelaar, K. Eikland and P. Notebaert
lp_solve, an open source (mixed-integer) linear programming system.
Homepage: http://lpsolve.sourceforge.net/5.5/.

[7] D. Bertsekas
The auction algorithm: A distributed relaxation method for the assignment problem.
Annals of Operations Research: Special Issue on Parallel Optimization, **14** (1988), 105–123.

[8] T. Cao, J. Hall and R. van de Geijn
Parallel Cholesky factorization of a block tridiagonal matrix.
International Conference on Parallel Processing Workshops (ICPPW'02), 2002, 327–335.

[9] H. Chen, T. Kirubarajan and Y. Bar-Shalom
Performance limits of track-to-track fusion vs. centralized estimation.
IEEE Transactions on Aerospace and Electronic Systems, **39**, 2 (Apr. 2003), 386–400.

[10] M. Chummun, T. Kirubarajan, K. Pattipati and Y. Bar-Shalom
Fast data association using multidimensional assignment with clustering.
IEEE Transactions on Aerospace and Electronic Systems, **37**, 3 (July 2001), 898–913.

[11] I. Cox and M. Miller
On finding ranked assignments with application to multitarget tracking and motion.
IEEE Transactions on Aerospace and Electronic Systems, **31**, 1 (Jan. 1995), 486–489.

[12] L. Kaplan and W. Blair
Assignment costs for multiple sensor track-to-track association.
In *Proceedings of the 7th International Conference on Information Fusion*, Stockholm, Sweden, June 2004.

[13] L. M. Kaplan, W. Blair and Y. Bar-Shalom
Simulation studies of multisensor track association and fusion methods.
In *Proceedings of the 2006 IEEE/AIAA Aerospace Conference*, Big Sky, MT, Mar. 2006.

[14] X. Li, Z. Luo, K. Wong and E. Bossé
An interior point linear pogramming approach to two-scan data association.
IEEE Transactions on Aerospace and Electronic Systems, **35**, 2 (Apr. 1999), 474–490.

[15] M. L. Miller, H. S. Stone and I. J. Cox
Optimizing Murty's ranked assignment method.
IEEE Transactions on Aerospace and Electronic Systems, **33**, 3 (July 1997), 851–862.

[16] K. Murty
An algorithm for ranking all the assignments in order of increasing cost.
Operations Research, **16** (1968), 682–687.

[17] K. R. Pattipati, S. Deb, Y. Bar-Shalom and R. B. Washburn
Passive Multisensor Data Association Using a New Relaxation Algorithm.
Multitarget-Multisensor Tracking: Advanced Applications, Artech House, 1990, ch. 7. Reprinted by YBS Publishing, 1998.

[18] K. R. Pattipati, R. Popp and T. Kirubarajan
Survey of assignment techniques for multitarget tracking.
In Y. Bar-Shalom and W. Dale Blair (Eds.), *Multitarget-Multisensor Tracking: Applications and Advances*, vol. III, Artech House, 2000.

[19] A. Poore and A. Robertson
A new class of Lagrangian relaxation based algortithms for a class of multidimensional assignment problems.
Computational Optimization and Applications, **8** (1997), 129–150.

[20] R. Popp, K. R. Pattipati and Y. Bar-Shalom
An *m*-Best 2-D assignment algorithm and multilevel parallelization.
Transactions on Aerospace and Electronic Systems, **35**, 4 (Oct. 1999), 1145–1160.

Javier Areta received the electrical engineer degree in 2001 from the University of La Plata, Argentina (UNLP). He is currently pursuing his Ph.D. in electrical and computer engineering at the University of Storrs, Connecticut.

His main interests lie in the areas of tracking, data fusion and signal processing.

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 Dept. 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 350 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 co-recipient 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 and the 2002 J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group.

Mark Levedahl B. Engineering (mechanical), McGill University, 1979.

Mr. Levedahl has over 25 years experience working in systems engineering, guidance, navigation, and control system development, battle management, multi-target multi-sensor tracking, and simulation. He has been employed by Raytheon since 1991 and working primarily on missile defense systems during that time. He is currently the Algorithm Team Leader for the Missile Defense National Team C2BMC program that is developing the integrated Command/Control and Battle Management system for world-wide ballistic missile defense. In this role, he has overall responsibility for the design and delivered performance of tracking and engagement management algorithms for the system. His work in missile defense has also included several interceptor systems including GBI-EKV, LEAP, MEADS, SM2-IVA programs, with focus on guidance methods, seeker requirements, and resulting system performance. In the 1980s, he was employed by Northrop designing and testing guidance, navigation, and control systems for unmanned air vehicles. Mr. Levedahl holds patents in Dynamic Wireless Utilization and in Sensor Data Association in the presence of biases, and is author or coauthor of a number of papers in multi-sensor tracking and systems interoperability. He is the primary author or a key contributor to a number of simulation systems with significant usage, including missile simulations at Northrop and Raytheon, and most recently the MDA sponsored BMD Benchmark used for missile defense.

Krishna R. Pattipati received the B.Tech. degree in electrical engineering with highest honors from the Indian Institute of Technology, Kharagpur, in 1975, and the M.S. and Ph.D. degrees in systems engineering from the University of Connecticut in 1977 and 1980, respectively.

From 1980–86 he was employed by ALPHATECH, Inc., Burlington, MA. Since 1986, he has been with the University of Connecticut, where he is a Professor of Electrical and Computer Engineering. His current research interests are in the areas of adaptive organizations for dynamic and uncertain environments, multi-user detection in wireless communications, signal processing and diagnosis techniques for complex system monitoring, and multi-object tracking. Dr. Pattipati has published over 300 articles, primarily in the application of systems theory and optimization (continuous and discrete) techniques to large-scale systems. He has served as a consultant to Alphatech, Inc. Aptima, Inc., and IBM Research and Development, and is a cofounder of Qualtech Systems, Inc., a small business specializing in intelligent diagnostic software tools.

Dr. Pattipati was selected by the IEEE Systems, Man, and Cybernetics Society as the Outstanding Young Engineer of 1984, and received the Centennial Key to the Future award. He was elected a Fellow of the IEEE in 1995 for his contributions to discrete-optimization algorithms for large-scale systems and team decision making. He has served as the Editor-in-Chief of the *IEEE Transactions on SMC: Part B—Cybernetics* during 1998–2001, vice-president for technical activities of the IEEE SMC Society (1998–1999), and as vice-president for conferences and meetings of the IEEE SMC Society (2000–2001). He was co-recipient of the Andrew P. Sage award for the Best SMC Transactions Paper for 1999, Barry Carlton award for the Best AES Transactions Paper for 2000, the 2002 NASA Space Act Award for "A Comprehensive Toolset for Model-based Health Monitoring and Diagnosis," the 2003 AAUP Research Excellence Award and the 2005 School of Engineering Teaching Excellence Award at the University of Connecticut. He also won the best technical paper awards at the 1985, 1990, 1994, 2002, 2004 and 2005 IEEE AUTOTEST Conferences, and at the 1997 and 2004 Command and Control Conferences.

Information for Authors

The Journal of Advances in Information Fusion (JAIF) is a semi-annual, peer-reviewed archival journal that publishes papers concerned with the various aspects of information fusion. The boundaries of acceptable subject matter have been intentionally left flexible so that the journal can follow the research activities to better meet the needs of the research community and members of the International Society of Information Fusion (ISIF).

Restrictions on Publication: The International Society of Information Fusion (ISIF) publishes in JAIF only original material that has not been either published or submitted for publication in an archival journal. However, prior publication of an abbreviated and/or preliminary form of the material in conference proceedings or digests shall not preclude publication in JAIF when notice is made at the time of submission and the previous publications are properly referenced in the manuscript. Notice of prior publication of an abbreviated version of the manuscript must be given at the time of submission to the transactions. Submission of the manuscript elsewhere is precluded unless it is refused for publication or withdrawn by the author. Concurrent submission to other publications and this journal is viewed as a serious breach of ethics and, if detected, the manuscript under consideration will be rejection.

Sanctions for Plagiarism: If a manuscript is substantially similar to any previous journal submission by any of the authors, its history and some explanation must be provided. If a manuscript was previously reviewed and rejected by a different journal, the authors shall provide to the area editor copies of all correspondence involving the earlier submission. In addition, the authors must discuss the reasons for resubmission and be prepared to deliver further material if more is requested. Submission of a previously rejected manuscript is acceptable as long as the motive for submission to JAIF is explained and the authors have addressed the feedback of the earlier reviewers. The authors of a manuscript that is found to have been plagiarized from others or contain a crossover of more than 25% with another journal manuscript by any of the authors will incur sanctions by the ISIF. The following sanction will be applied for any of the above-noted infractions: (1) immediate rejection of the manuscript in question; (2) immediate withdrawal of all other submitted manuscripts by any of the authors to any of the ISIF publications (journals, conferences, workshops); (3) prohibition against all of the authors for any new submissions to ISIF publications, either individually, in combination with the authors of the plagiarizing manuscript as well as in combination with new coauthors. The prohibition shall continue for at least two years from notice of suspension.

Types of Contributions: Contributions may be in the form of regular papers or correspondence. The distinction between regular papers and correspondence is not one of quality, but of nature. Regular papers are to be a well-rounded treatment of a problem area, while a correspondence item makes one or two points concisely. In regular papers, the title, abstract, and introduction should be sufficiently informative to illuminate the essence of the manuscript to the broadest possible audience and to place the contributions in context with related work. The body of the manuscript should be understandable without undue effort by its intended audience. Correspondence should be less discursive but equally lucid. Authors should be aware that a well-written correspondence are usually published more rapidly than a regular paper.

Submission of Manuscript for Review: Manuscripts are submitted electronically via the internet at http://jaif.msubmit.net. For peer review, manuscripts should be formatted in a single column and double spaced.

General Information for JAIF Authors and Submission of Final Manuscript: General information for JAIF authors is available at http://www.isif.org/jaif.htm, and specific document preparation information for final manuscript for publication is given below and can be found at http://www.isif.org/JAIF Manuscript Preparation1.doc.

Copyright: It is the policy of the ISIF to own the copyright to the technical contributions it publishes; authors are required to sign an ISIF copyright transfer form before publication. This form appears in JAIF from time to time and is available on the web site listed above. If a copyright form is not submitted with the manuscript, it will be requested upon contribution acceptance. Publication will not take place without a completed copyright form.

Page Charges: Since ISIF has elected to support the publication of JAIF with the revenues of conferences and membership dues, page charges are not mandated for publication in JAIF. Page charges for journals of professional societies are traditional, and at some time in the future, page charges may be introduced to cover the expenses associated with the publication of JAIF. Payment of page charges will not be a prerequisite for publication.

Discloseability: The ISIF must, of necessity, assume that material submitted for publication is properly available for general dissemination to the audiences for which JAIF serves. It is the responsibility of the author, not ISIF, to determine whether disclosure of materials requires prior consent of other parties, and, if so, obtain it.

Preparation of Manuscript for Publication: After a manuscript is recommended for publication by the editorial staff, the corresponding author prepares the manuscript according to the guidelines below and submits the final version of the manuscript as a pdf file and the supporting manuscript files. Before prepar-

ing their manuscript for publication, authors should check for the more update version of the guidelines at http://jaif.msubmit.net.

All manuscripts must be submitted electronically in pdf format with the supporting files in one of the following formats: Plain TeX, AMS-TeX, LaTeX, or MS-Word with the MathType extension for equations and in-text mathematical material. The pdf file must be created with graphics resolution set to 300 dpi or greater. The author must approve the pdf file because it will be used as the basis for rectifying any inconsistencies in the supporting files. The supporting files must be prepared according to the following guidelines.

- Text files should include title, authors, affiliations, addresses, a footnote giving the dates for submission and revision and name of the editor that handed the review, and an optional footnote acknowledging a sponsor for the research. The information must be included in the text files in this order.
- Abstract must be included. The abstract must include the key words for the manuscript and include no more than 300 words for a regular paper and 150 words for a correspondence.
- Authors should number main section headings as 1, 2, 3, etc., and subsections as 1.1, 1.2, or 2.1. All headings should be typed with title format (i.e., first letter caps and lower case)—not in ALL CAPS. The typesetter will convert to all caps in final formatting as required.
- Authors should format references very carefully according to the examples given below. Style for authors' names are initials followed by last name and it must be followed precisely. References must be listed alphabetically by (i) last name of first author, (ii) initials of first author, (iii) last name of second author, (iv) initials of second author, etc. For manuscripts with common authors and order of authors, the date of publication should be used to select order with the earlier publications being list first. The names of publications must be spelled completely. In the references, the author should use only approved abbreviations that include: No., Vol., p., pp., AIAA, IEEE, IEE, and SPIE.
- Authors who use one of the TeX variants must provide a list of their macros and the files for the macros. Authors who use LaTeX must include the bbl and aux files.
- All figures must be submitted electronically as HIGH resolution (300 dpi or better) in Color graphics or GRAYSCALE (where shading is involved) or BLACK AND WHITE (if simple line art) graphics files (tif, eps, jpg, bmp). Each figure must be supplied as a separate graphics file. Graphics (or captions) should NOT be embedded in the text files. The figures must be included in the pdf file of the full article and the pdf file must be created with graphics resolution set to 300 dpi or greater.
- A separate file including all figure captions must be included.
- Each table must be submitted in a separate file. Tables (or captions) should NOT be included in text files and should be in form of DATA—rather than graphics—files.
- A separate file including all table captions must be included.
- A separate text of the biography of each author must be submitted. The text file should be less than 500 words.
- Separate graphics files of each author's photo should be provided as a grayscale graphics file or a color graphics file.

Examples of the references are alphabetized correctly and listed below.

BOOK:

[1] R. E. Blahut
Theory and Practice of Error Control Codes. Reading, MA: Addison-Wesley, 1983.

PROCEEDINGS ARTICLE:

[2] T. Fichna, M. Gartner, F. Gliem, and F. Rombeck
Fault-tolerance of spacebome semiconductor mass memories.
In Twenty-Eighth Annual International Symposium on Fault-Tolerant Computing, Digest of Papers, 1998, 408–413.

BOOK:

[3] P. K. Lala
Fault Tolerant and Fault Testable Hardware Design. Englewood Cliffs, NJ: Prentice-Hall, 1985.

WEB SITE:

[4] National Semiconductors Inc.
Homepage: http://www.national.com.

PROCEEDINGS ARTICLE:

[5] C. Paar and M. Rosner
Comparison of arithmetic architectures for reed-solomon decoders in reconfigurable hardware.
In *Proceedings of the Symposium on Field-Programmable Custom Computing Machines*, Apr. 1997, 219–225.

JOURNAL ARTICLE:

[6] N. R. Saxena and E. J. McCluskey
Parallel signature analysis design with bounds on aliasing.
IEEE Transactions on Computers, **46**, 4 (Apr. 1997), 425–438.

[7] C. I. Underwood and M. K. Oldfield
Observations on the reliability of cots-device-based solid state data recorders operating in low-Earth orbit.
IEEE Transactions on Nuclear Science, **47**, 4 (June 2000), 647–653.

WEB SITE:

[8] Xilinx Inc.
Homepage: http://www.xilinx.com.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

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

BOARD OF DIRECTORS*

2004–2006	2005–2007	2006–2008
Erik Blasch	Per Svensson	David Hall
Jean Dezert	Darko Musicki	Ivan Kadar
Xiao-Rong Li	Eloi Bossé	Elisa Shahbazian

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

PAST PRESIDENTS

Pierre Valin, 2006	Xiao-Rong Li, 2003	Yaakov Bar-Shalom, 2000
W. Dale Blair, 2005	Yaakov Bar-Shalom, 2002	Jim Llinas, 1999
Chee Chong, 2004	Parmod Varshney, 2001	Jim Llinas, 1998

SOCIETY VISION

The International Society of Information Fusion (ISIF) seeks to be 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 dependant 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.

Additional Copies

If you need additional copies, please contact Tammy Blair at 404-407-7726 or tammy.blair@gtri.gatech.edu for pricing information.

Journal of Advances in Information Fusion Volume 1 Number 2 December 2006 ISSN 1557-6418