June 2009 Volume 4 Number 1 ISSN 1557-6418

JAIF 09

Journal of Advances in Information Fusion

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

Regular Papers | Page

From the Editor-In-Chief

The Critical and Key Element of a Peer Review...

INTERNATIONAL SOCIETY OF INFORMATION FUSION

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

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

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

From the Editor in Chief:

June 2009

JAIF Editorial Board

The Board of Directors of ISIF voted in 2004 to establish JAIF as a peer reviewed journal publishing articles in the area of information fusion. The critical and key element of a peer review journal is its editorial board. As Editor-in-Chief (EIC) for JAIF, I have attempted to assemble a strong editorial board with international representation from both academia and industry. The editorial board includes 15 editors from seven different countries and six editors from academia and nine editors from industry.

The editorial board for JAIF includes an EIC, an associate EIC, administrative editors, area editors, and associate editors. The inside cover of each issues identifies the individuals that hold these positions and the list of the technical areas represents the scope of JAIF.

The EIC is responsible for the day-to-day editorial operations of the JAIF. The EIC is responsible for identifying and maintaining the appropriate technical areas of JAIF. Editorial operations primarily relate to the timely review of manuscripts submitted to JAIF. The EIC consults with the vice president of ISIF for publications (VP-Pubs) on extraordinary issues. The EIC and the VP-Pubs consult on the strategic vision for JAIF. For 2005–2010, Yaakov Bar-Shalom served and continues to serve as the VP-Pubs.

The associate EIC serves in roles of the EIC as designated by the EIC. The EIC consults with the associate EIC on strategic and management issues. Generally, an associate EIC will also be an Area Editor for JAIF.

The administrative editor is responsible for manuscript post-acceptance handling and subsequent publication. This includes management of copy-editing, typesetting, assembly of complete issues of JAIF, and delivery to ISIF of properly meta-tagged and indexed manuscripts.

Each technical area of JAIF is managed by an area editor, who is directly responsible for all manuscripts submitted to that area. A manuscript is assigned to an associate editor (AE) who directly manages its review. Area editors will also serve in the role of an AE in their area of expertise and can assign manuscripts to themselves for review management.

Each manuscript submitted to the JAIF is assigned to an AE. The AE assigns reviewers and assures that these referees deliver their reviews in a timely fashion. The AE delivers the ultimate decision (accept/reject/revise) on each manuscript. Unless the AE is also the area editor, the AE is responsible to the area editor of their technical area.

Area editors are appointed by the EIC. In consultation with the associated area editors, AEs are appointed by the EIC. While the EIC has ultimate authority over the selection of AEs and area editors, it is expected that in most cases that the VP-Pubs will be consulted on appointments by the EIC. Appointments are made based on publication record, familiarity with the subject area, prior relationship and contribution to the ISIF community, and recommendations from relevant experts. Publication record must include multiple articles in peer reviewed journals with publication standards similar to JAIF. Familiarity with the subject area and prior relationship to the ISIF community is best demonstrated by quality reviews of articles for JAIF. Each editor's performance (acceptance rate, timeliness, etc.) is continuously reviewed by the area editor and EIC. The EIC is appointed by the VP-Pubs and serves at the pleasure of ISIF BoD—normally for a term of six years, beginning in January.

Further details on the editorial policies and procedures of JAIF are available for review at www.isif.org. These peer review guidelines establish the peer review standards for JAIF and it is the intent of the editorial board to maintain those standards.

William Dale Blair
Editor in Chief

Application of Intent Inference for Air Defense and Conformance Monitoring

PEK HUI FOO
GEE WAH NG
KHIN HUA NG
RONG YANG

Intent inference involves the analysis of actions and activities of a target of interest to deduce its purpose. This paper proposes an approach for intent inference based on aircraft flight profile analysis. Simulation tests are carried out on flight profiles generated using different combinations of flight parameters. In each simulation test, Interacting Multiple Model-based state estimation is carried out to update the state vectors of the aircraft being monitored. Relevant variables of the filtered flight trajectory are subsequently used as inputs for a Mamdani-type fuzzy inference system. Research on two applications is reported. The first application involves the determination of the likelihood of weapon delivery by an attack aircraft under military surveillance. Test results verify that the method is feasible and is able to provide timely inference. By extending the method to take the environmental context of the tracked aircraft into consideration when executing the inference process, it is likely that the military defenders would be able to raise their alert earlier against potential adversaries. This would provide them with more time to react and devise pre-emptive counteraction. The second application concerns conformance monitoring in air traffic control systems. Experimental results show that the proposed solution can be used to assist air traffic control system operators in determining if aircraft navigate according to planned trajectories. Consequently, corrective action can be taken on detection of anomalous behavior. A brief discussion on extending the proposed method to deal with multiple aircraft is also presented.

Manuscript received September 16, 2007; released for publication October 13, 2008.

Refereeing of this contribution was handled by Pierre Valin.

Authors' address: DSO National Laboratories, 20 Science Park Drive, Singapore 118230, Singapore.

1557-6418/09/$17.00 © 2009 JAIF

1. INTRODUCTION

The human brain has remarkable capabilities in perception and reasoning. However, the amount of complex data/information that can be processed by the human brain is constrained by the limited memory capacity. Hence, computational tools are necessary to provide cognitive aid to the human brain in attaining better performance in intellectual tasks, such as decision making.

Intent inference is about analyzing the actions and activities of an opponent or a target of interest to obtain a conclusion (prediction) on its purpose [3, 10, 18, 24]. Generally, data (collectively called observables) concerning the opponent are first collected from available sources. Next, the data are fused to obtain useful information. Finally, the fused information is utilized to derive the inferred intent of the opponent. It is desirable that intent inference be able to provide three kinds of hypotheses about an opponent's objective [3, 18]:

- Descriptive intent inference—provides insight into the motivations behind preceding actions;
- Predictive intent inference—anticipates the opponent's future actions given his deduced goals;
- Diagnostic intent inference—detects differences between predicted and observed actions to reveal possible errors.

Accurate prediction of an opponent's intention, actions and reactions would be useful for the purpose of devising effective responses to his actions, as well as planning for one's own operations.

Intent inference has been used in applications such as intelligent transportation systems (infer and detect a driver's intent [36]) and air traffic management (ATM) (predict the future trajectory of an air vehicle and the states of nearby aircraft [20, 42]). Other applications include the medical domain, recommender systems, tutoring systems and team intent identification [18].

In this paper, we report our research on two applications of intent inference [9, 25]. The first task is to determine the intent of the pilot (equivalently, the flight mission) of an aircraft being tracked by a military surveillance system [25]. The second involves conformance monitoring in air traffic control (ATC) systems [31].

This paper is organized as follows. Section 2 provides a general discussion on intent inference and a brief review on related work from the research literature. Section 3 describes our proposed fuzzy logic approach for intent inference based on the analysis of flight profiles for attack aircraft. In addition, the environmental context of the tracked aircraft is taken into consideration during the execution of the inference process. The impact of this additional factor on the inference outcome is investigated. Four different test scenarios are used to evaluate the feasibility of the proposed method. Section 4 is focussed on conformance monitoring in ATC/ATM systems. Section 5 presents simulation tests and results.

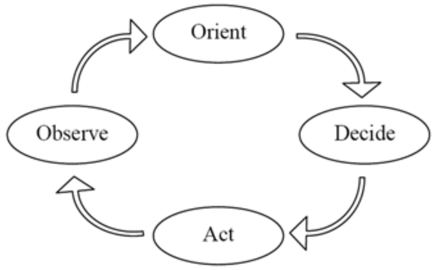


Fig. 1. The OODA Loop.

Section 6 gives a discussion on handling an approach by multiple aircraft. Section 7 provides a summary on this paper.

2. INTENT INFERENCE

The *Boyd Control Loop* (also called *Boyd's Decision Loop* or the *Observe, Orient, Decide, and Act* (OODA) *Loop*) [11, 27] is a popular model that has been used for formalizing concepts of tactical command and decision making. It describes human and organizational behavior as a continuous, iterative and cyclic process of *Observation* (represents event perception), *Orientation* (corresponds to the process of memory and cognition, the activity that provides environmental context and individual expectations), *Decision* (describes the process of cognitive comparison) and *Action* (equals the resulting behavior). In particular, the function Orientation shapes the way the other functions, Observation, Decision and Action, are done.

The emphasis of this model is placed on shortening the cycle to perform the Observe to Act loop (see Fig. 1):

- Observe—gather data from the environment via human and related senses,
- Orient—gain situation awareness and perform situation and impact/threat assessment based on the information derived from the data obtained,
- Decide—respond to situation and work out follow-up actions,
- Act—execute the planned response,

to the extent that the opponent cannot respond in time to carry out countermeasures, thus gaining superiority in the engagement. The OODA Loop can also be applied to computer-assisted cognition.

An intent inference system provides reasoning about the opponent's intent, mission objective, or motivation. By nature of the inference mechanism, the intent inference system will also be able to provide prediction on the opponent's possible future actions or activity according to the inferred intent. Thus, it serves as useful decision support to the decision maker. In this way, the inference system not only contributes to better situation awareness and aids in resolving ambiguity that arises from multi-source fusion, but further assists the decision maker in his cognitive task and helps in shortening the decision making process.

Intent inference is a relatively young and challenging research area as compared to the maturing lower level data fusion. Emerging interest in the application of this research area can be found in the military arena [3, 37] and antiterrorism [12, 16]. Generally, intent and activity inference requires a cognitive architecture with knowledge-based modeling. Inputs to the inference system are information gathered through intelligent autonomous agents or provided by multiple sensing sources, including reports from human intelligence. Through modeling, the structure and pattern of opponent entities, as well as their behavior and relationships, are captured. The focus of the inference mechanism is on contextual and relational reasoning as opposed to single entity reasoning at lower level fusion processes. The inference mechanism may be based on a rule-based system or a more dynamic reasoning system such as Bayesian networks. In this paper, a fuzzy inference system (FIS), also known as a fuzzy-rule-based system, is used.

2.1. Related Research Work

A method for pilot intent inference in real-time was investigated in [19]. It was based on plausible models of intent and a process for identifying models that matched observed aircraft motion best. The models were ranked based on their correlation with measured aircraft motion. The highest ranked plausible models of intent made up the best estimate of the aircraft intent. Sequences of actions were executed to infer guidance and navigation task intents of the tracked aircraft. The inferred intent was then used as a basis for trajectory prediction.

The authors of [41] proposed an intent-based trajectory prediction algorithm to carry out maneuvering aircraft tracking, aircraft intent inference and trajectory prediction. A hybrid estimation algorithm was used for estimating the states and flight mode of the aircraft. Intent inference was posed as a maximum likelihood problem. Pilot intent inference was obtained via the combination of the state and flight mode estimates, air traffic control regulations, the flight plan of the aircraft and environment information. The inferred intent and the aircraft motion (state and flight mode estimates) were used for the computation of trajectory prediction. The proposed algorithm was tested and analyzed through simulations in different scenarios representative of aircraft operations.

In [1], a hybrid system model of intent inference was constructed for air traffic controllers. An algorithm based on the Interacting Multiple Model (IMM) Kalman filter (the State Dependent Transition Hybrid Estimation algorithm) was implemented for state estimation, as well as the generation of residuals (discrepancies) between the observed aircraft states and the expected aircraft states. The residual mean was generated based on probabilistic methods. The proposed model was applied to an example problem on conformance monitoring. A statistical test was carried out on the residual means for both the conformance monitoring model and the actual aircraft system to obtain a conclusion/decision on conformance or non-conformance.

Conformance monitoring in air traffic control is a relatively new application of intent inference. Some research work based on fault detection has been done in this area [30–35] and will be discussed in Section 4.

2.2. Inference Mechanism

Classification is the process of inferring the concept behind an available collection of observations. This task covers any context in which some decision or forecast is made based on available information. It involves the establishment of a mapping from a measurement (an observation) space to a decision space. Input measurement/observation data is assigned into one or more predetermined classes based on the selection/extraction, as well as the processing or analysis, of significant features or attributes. Some commonly used approaches to classification are briefly discussed below [17].

2.2.1. Statistical Approach

Statistical (or decision theoretic) classifiers are generally characterized as having an explicit underlying probabilistic model. In a parametric classification procedure, a set of characteristic measurements (features) are extracted from the input data, and are used to assign each feature vector to one of the predetermined classes. Features are assumed to be generated by a state of nature, the underlying model represents a state of nature, set of probabilities, or probability density functions, that are conditional on the classes.

There are cases when there is insufficient prior information available, or when it is not necessary, to make assumptions about the distribution associated with the feature vector in the different classes. Under such circumstances, it is possible to use non-parametric estimation of the pdf involved to build distribution-free methods of classification (that is, non-parametric classifiers).

Statistical classifiers generally work reasonably well for problems in which structures are not deemed significant.

2.2.2. Neural Network Approach

A neural network assumes that a set of input data and their correct classifications are given. The architecture of a neural net includes layers of interconnected nodes. It is characterized by a set of weights and activation functions which determine the transmission of information from the input layer to the output layer. The training data is used to train the neural network and adjust the weights until the correct classifications are obtained. The complete network generally represents a complex set of interdependencies, which may incorporate an arbitrary degree of nonlinearity.

Neural networks are suitable for solving problems with a large amount of features and classes. They can be applied to problems that involve generalization, parallel processing, or discrimination among classes with highly nonlinear boundaries.

2.2.3. Fuzzy Logic Approach

Classification is often done with some degree of uncertainty. In problems with data that are noisy and distorted, complications can arise and lead to ambiguous situations in which classified data may belong in some degree to more than one class, or the classification outcome itself may be in doubt. Fuzzy logic (or fuzzy set theory) can be introduced to deal with such problems. In fuzzy classification, an input data entity is assigned a membership value in the interval $[0, 1]$ in each predetermined class.

2.3. Proposed Approach

We propose that intent inference be carried out via a fuzzy logic approach (conceptual information on fuzzy logic used in this paper [15, 38, 39] is given in the Appendix). The main reasons that motivate the use of the proposed approach are as follows.

Firstly, compared to statistical and probabilistic methods used in most related research work, fuzzy logic techniques are particularly suitable for modeling problems with inherent imprecision properties [11, 23]. The problems to be discussed in this paper involve observation/information associated with human cognitive processes such as thinking and reasoning, in which uncertainties and imprecision are usually inherent. Therefore, it is appropriate to use fuzzy logic to deal with these problems.

Secondly, fuzzy logic techniques are useful for the fusion of information from multiple input sources and the application of heuristics to determine the overall status of the inputs [7]. Hence, for each problem in this paper, the information obtained from tracking the subject aircraft can be fused to determine the pilot intent, which is required by the surveillance/monitoring system users concerned for decision making.

Thirdly, implementation of fuzzy logic is simple, fast and efficient [21, 38]. This would be useful for problems in which computational load/time is a critical factor, such as the two problems of interest here. For the first task on air defense, it is essential to take preemptive action against potential adversaries as quickly as possible, in order to avert possible attacks. For the second problem on conformance monitoring in air traffic control systems, it is important to minimize the delay in correcting any deviant aircraft behavior that is detected.

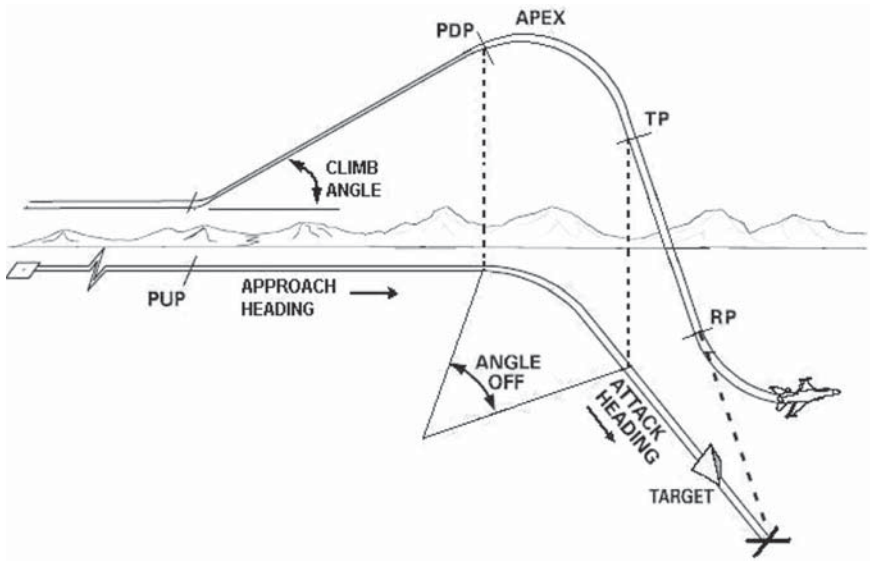


Fig. 2. Flight profile for offset pop-up delivery.

A fuzzy inference system is a computing framework based on the concepts of fuzzy set theory, fuzzy rules and fuzzy reasoning (an inference procedure which derives conclusions from a set of fuzzy rules and available information) [15]. The basic structure of a fuzzy inference system comprises three conceptual components:

- rule base—contains a selection of fuzzy rules,
- database—defines the membership functions used in the fuzzy rules,
- reasoning mechanism—performs the inference procedure upon the rules and known facts to derive a reasonable output or conclusion.

The inference mechanism used in this paper is based on the widely accepted Mamdani's fuzzy inference method [15], which was one of the first control systems built using fuzzy set theory. It was proposed as an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators.

The Mamdani-type FIS used here is generated using the MATLAB Fuzzy Logic Toolbox [38, 39]. The fuzzy inference process has five parts, namely, fuzzification of the input variables, application of the fuzzy operator in the antecedent, implication from the antecedent to the consequent, aggregation of the consequents across the rules, and defuzzification. Details on each part of the fuzzy inference process implemented for the two applications discussed in this paper are provided in Sections 3 and 4.

3. WEAPON DELIVERY BY ATTACK AIRCRAFT

Effective intent inference will greatly enhance the defense capability of a military force in taking preemptive action against potential adversaries. It serves as a form of advance warning in the prevention of a crisis (for instance, enemy attack) or facilitates the moderation of the impact of such a crisis. For an air defense system, the ability to accurately infer the likelihood of a weapon delivery by an attack aircraft is critical.

The type of weapon delivery for attack aircraft considered in this paper is offset pop-up delivery. The definitions for some terms pertaining to this form of weapon delivery are stated below. Section 3.1 provides a brief description of offset pop-up delivery [40].

- Pop Point (PUP)—a position at which the pop-up attack is initiated, the point where climb is initiated.
- Pull-Down Point (PDP)—a maneuver point where one transitions from the climbing to the diving portion of a pop-up delivery.
- Apex—the highest altitude in the pop-up delivery profile.
- Track Point (TP)—the starting point of tracking prior to arriving at planned release altitude.
- Release Point (RP)—the point at which weapon is released.

A tracked aircraft is considered to have constant speed, with the velocity components in the horizontal plane (parallel to ground) and the vertical axis (parallel to altitude) varying in different phases of the trajectory. In this application, altitude, distance and velocity are measured in feet above ground level (AGL), feet and knots respectively, unless otherwise stated.

3.1. Typical Offset Pop-up

The pop-up approach heading, as shown in Fig. 2 [8], is at an angle (varies with the planned climb angle) from 15° to 90° from the final attack heading. This allows the pilot to acquire the target as soon as possible and maintain visual contact until weapon delivery is completed.

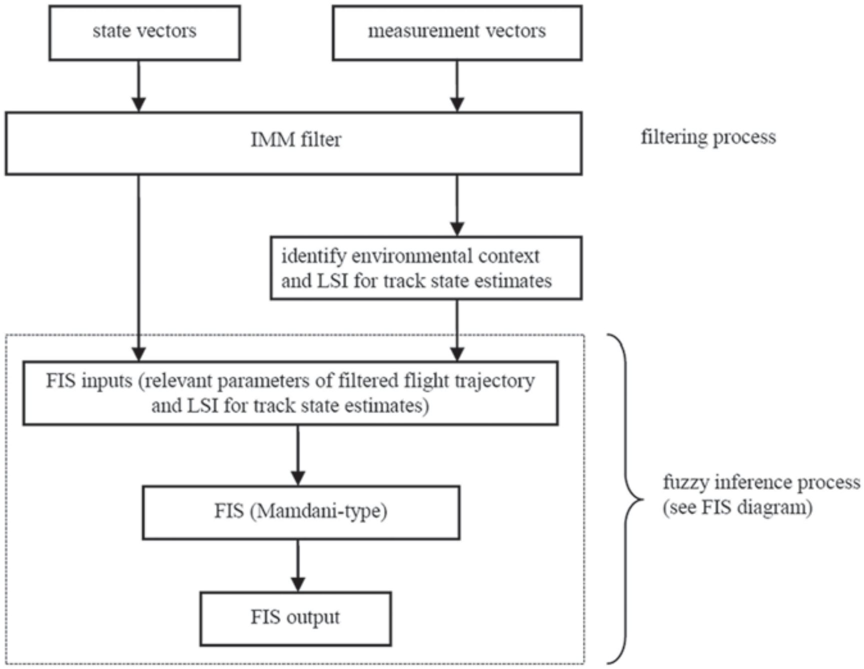


Fig. 3. Overview of proposed system.

The pilot initiates the pop-up over a preplanned pop point at a minimum airspeed of 450 knots calibrated airspeed (KCAS). He selects his desired power, makes a 3–4 G wings-level pull to the desired climb angle and initiates a chaff/flare program. After popping, he has to maintain the planned climb angle and monitor the altitude gained.

When approaching the preplanned pull-down altitude, the pilot makes an unloaded[1] roll in the direction of the target. He then performs a 3–5 G pull-down to intercept the planned dive angle. Interception of the planned dive angle while pointed at the aim-off point is a critical factor in attaining preplanned delivery parameters. It is usually acceptable to have minor deviations in the attack heading.

During the maneuver, corrections are made to compensate for minor errors in the pop point or unexpected winds in the climb to the apex at the planned altitude. The planned apex altitude is normally achieved about half way through the pull-down maneuver.

For safety reasons, a pilot would most probably abort a pop-up attack immediately if at least one of the following conditions arises:

- the actual dive angle exceeds the planned one by more than 5°,
- the airspeed goes below 350 KCAS (300 KCAS above 10000 feet AGL).

The occurrence of such conditions would result in inaccuracy in the impact point of the released weapon.

3.2. Process and Techniques

Our proposed procedure for inferring the possibility of weapon delivery by a tracked attack aircraft, based on flight profiles, is given below.

Procedure 1

1. For an aircraft being tracked, record its state information (sensor measurement data) through observation.
2. Apply the IMM algorithm [22, 24] to update the track state estimates.
3. For each track state estimate, use the position components to identify the environmental context and hence the corresponding location sensitivity index (LSI) (details in Sections 3.2.1 and 5.1).
4. Fuzzy inference process
   a. Input
      i. relevant parameters of the filtered flight trajectory, and
      ii. LSI obtained in Step 3,

      to a Mamdani-type fuzzy inference system gener ated using the MATLAB Fuzzy Logic Toolbox [38, 39].
   b. Output produced by the FIS is the inferred possibility of weapon delivery by the tracked aircraft.

[1]In aeronautics, the *lift* on an aircraft is the component of total air force acting on the aircraft which is perpendicular to the direction of flight and is normally executed in an upward direction. The *load factor* is the ratio of the lift on an aircraft to the weight of the aircraft, which is expressed in multiples of *G*, with 1 G representing conditions in straight and level flight.

Unloaded: the situation in which the load factor is 0 G, where every occupant of an aircraft experiences a feeling of weightlessness.

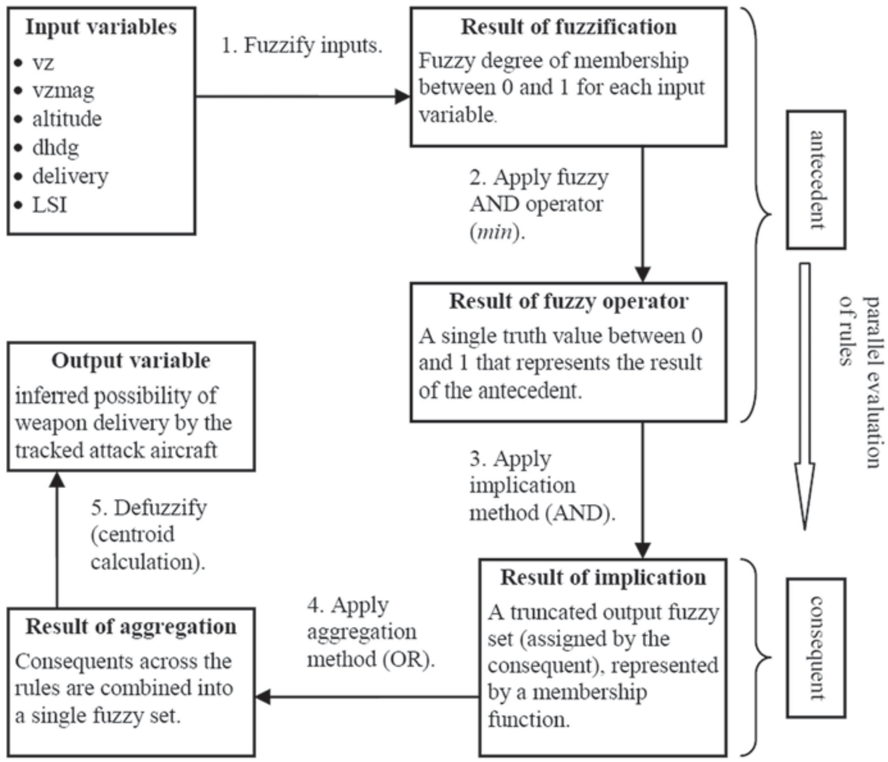


Fig. 4. Fuzzy inference system.

An overview of the system for the proposed approach is shown in Fig. 3. The entire fuzzy inference process is shown in Fig. 4. The following subsections provide details on the fuzzy inference process.

3.2.1. Fuzzification of the Input Variables

In the first step, each input variable is a crisp/non-fuzzy numerical value within its universe of discourse and is assigned a linguistic value in the interval $[0,1]$ via a membership function. The input variables considered in the current application are obtained from kinematic parameters of the filtered flight trajectory. Elaboration on each of the input variables, with respect to the tracked aircraft, is given below.

The first variable is the velocity along the vertical axis (abbreviated *vz*). It is classified as either positive (denoted by ">0") or negative (denoted by "<0"), indicating either upward or downward motion respectively. The second variable is the magnitude of vz (abbreviated *vzmag*). The third variable is the altitude. The fourth variable is an indicator for the occurrence of a change in heading (measured in radians, abbreviated *dhdg*) during the time interval between consecutive scans. A change in heading is considered to have occurred when the difference in heading between two consecutive records along the filtered flight trajectory exceeds a chosen threshold value ($\pi/180$ radians in the current application). The fifth variable is an indicator for the likelihood of a weapon delivery (abbreviated *delivery*) by the tracked aircraft. A weapon delivery is considered unlikely when at least one of the following conditions occurs:

- the actual dive angle exceeds the planned one by more than 5°,
- the airspeed goes below 350 KCAS (300 KCAS above 10000 feet AGL).

The sixth variable is an index representation of the environmental context of the tracked aircraft, named *location sensitivity index* (abbreviated *LSI*). The LSI is based on the degree of sensitivity of the spatial domain in which the tracked aircraft is traveling. High LSI corresponds to highly sensitive locations, including vicinities of critical infrastructure such as government establishments. Low LSI corresponds to locations with low sensitivity, including regions that are remote or not habitable.

Figs. 5 to 10 show the membership functions for the six input variables. Table I shows the symbols and their corresponding linguistic values for membership functions (where applicable).

TABLE I
Symbols used for Membership Functions

Symbol	VL	L	M	H	VH
Linguistic value	Very Low	Low	Medium	High	Very High

The number of levels for the linguistic values for membership functions can vary according to the amount

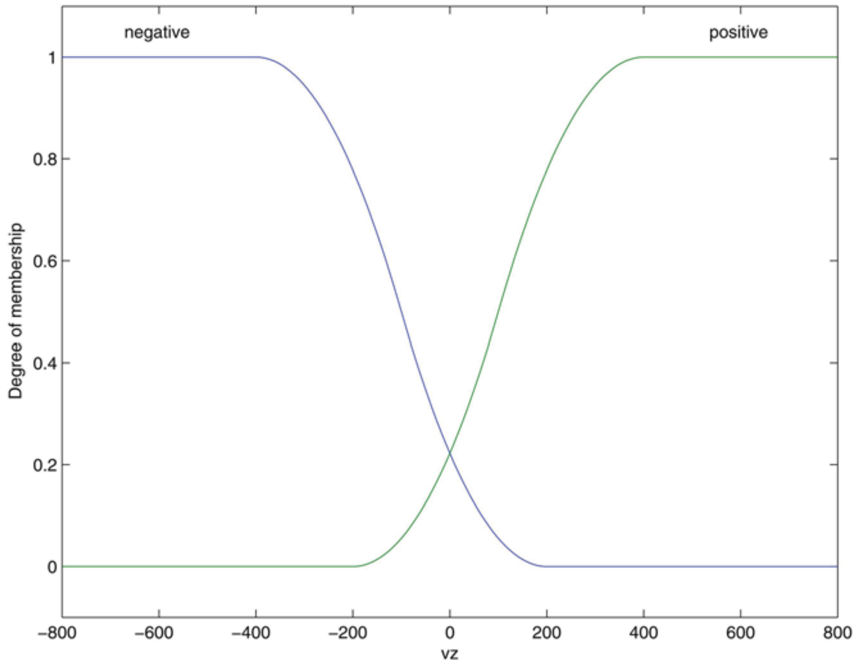


Fig. 5. Membership functions of "vz."

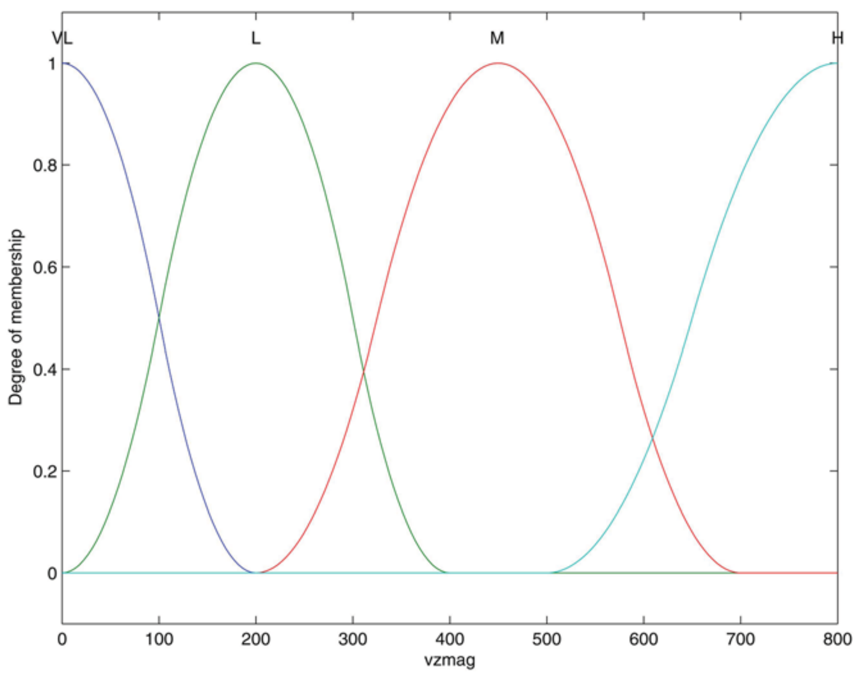


Fig. 6. Membership functions of "vzmag."

of information available. Labels that are more descriptive can be used for various levels of linguistic values of a variable. An example is to use words such as fast, slow and constant when labeling different degrees of membership for variables related to velocity/speed.

3.2.2. Application of Fuzzy Operators

After fuzzification of the inputs, the degree to which each part of the antecedent is satisfied for each rule is known. When an antecedent of a given rule has multiple parts, a fuzzy operator (such as those defined in the Appendix) has to be applied to the multiple membership values from fuzzified input variables, in order to obtain one single truth value. This output value (which lies in $[0,1]$) represents the result of that antecedent for that rule and will be applied to the output function.

3.2.3. Application of Implication Method

For each rule, apply a weight (1 is used in this paper) to the single truth value given by the antecedent. Then implement the implication on this weighted value using

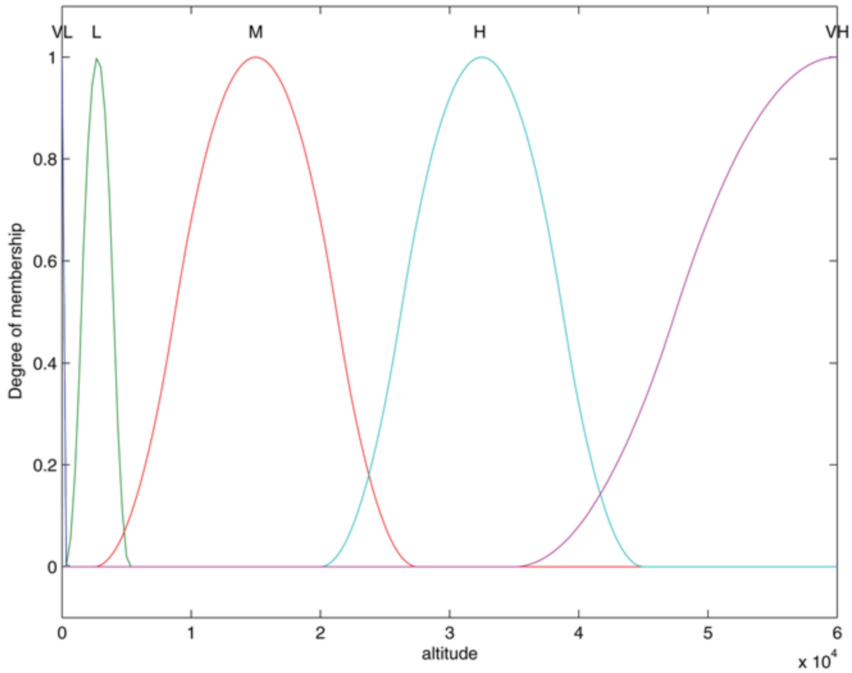


Fig. 7. Membership functions of "altitude."

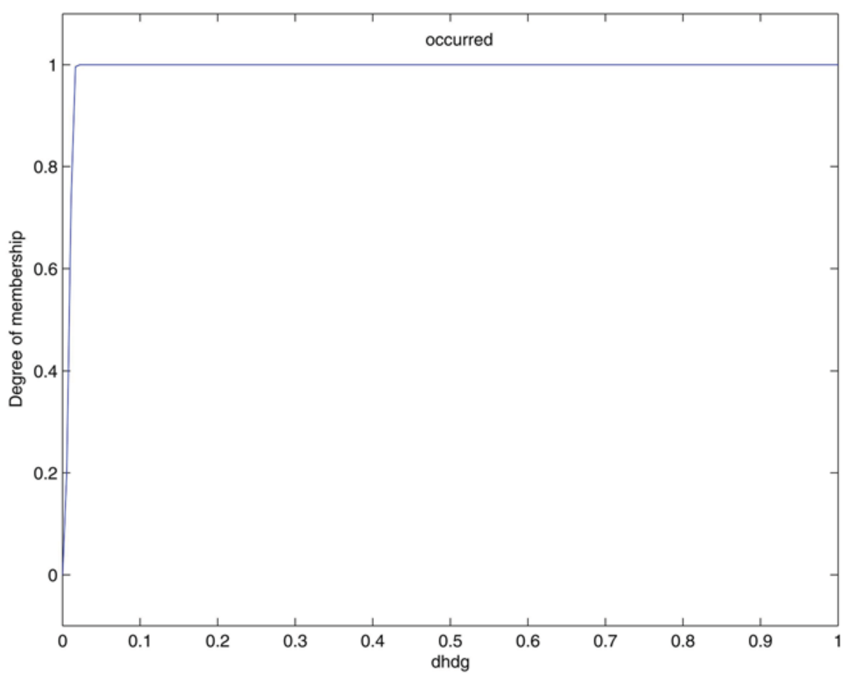


Fig. 8. Membership function of "dhdg."

the built-in AND method: min (minimum) function [38, 39]. The implication process yields an output fuzzy set (assigned by the consequent) which is truncated to the level of the weighted truth value of the antecedent. The rules used in the current application are listed in Table II. They are based on the expected characteristics of the motion along an offset pop-up delivery profile.

Fig. 11 shows the membership functions for the output variable (inferred possibility of weapon delivery by the tracked attack aircraft, abbreviated *pos*). The complexity of the rules can be modified according to the amount of information available.

3.2.4. Aggregation of All Outputs

It is necessary to determine an approach to combine the rules in a fuzzy inference system in order to reach a decision/conclusion. The output fuzzy sets of each rule (obtained via the preceding implication method) are unified to form a single output fuzzy set, whose membership function assigns a weighting for every

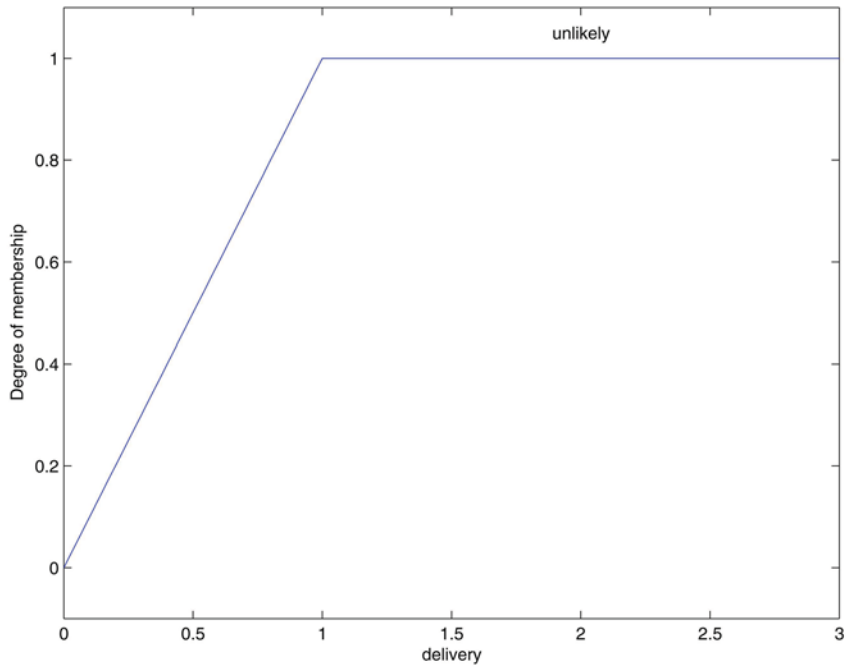


Fig. 9. Membership function of "delivery."

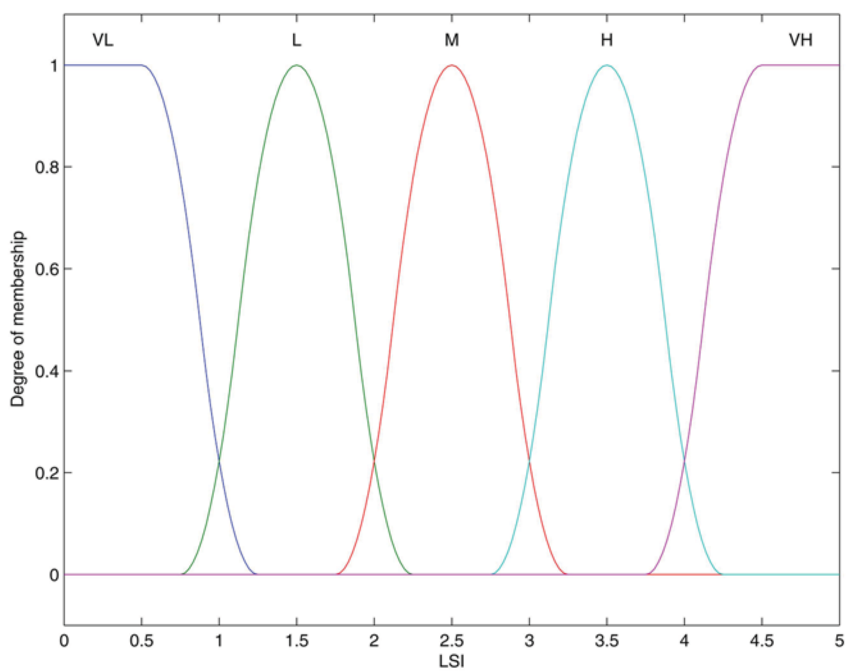


Fig. 10. Membership functions of "LSI."

output value. The aggregation process inputs are the truncated output membership functions returned by the preceding implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable. This paper utilizes the built-in OR method: max (maximum) function [38, 39] for the aggregation process. Therefore, the final membership function value is given by the maximum value among the consequent membership function values for each of the rules in the fuzzy inference system.

3.2.5. Defuzzification

In the last step of the fuzzy inference process, let F denote the output fuzzy set of the preceding aggregation process and Z denote the universe of discourse that F is in. Let $\mu_F(\cdot)$ be the aggregated output membership function representing F. Defuzzification of F yields the output of the fuzzy inference system, which is a single crisp/non-fuzzy number [15]. The built-in method of centroid calculation [38, 39] is used in this paper. The defuzzified output, z_{COA}, is the center of area under

TABLE II
Rules for Fuzzy Inference System (Weapon Delivery by Attack Aircraft)

R1.	(altitude is VL) $\rightarrow$ (pos is VL).
R2.	($vz > 0$) & (dhdg is NOT occurred) & (LSI is VL) $\rightarrow$ (pos is L).
R3.	($vz > 0$) & (dhdg is NOT occurred) & (LSI is L) $\rightarrow$ (pos is L).
R4.	($vz > 0$) & (dhdg is NOT occurred) & (LSI is M) $\rightarrow$ (pos is M).
R5.	($vz > 0$) & (dhdg is NOT occurred) & (LSI is H) $\rightarrow$ (pos is M).
R6.	($vz > 0$) & (dhdg is NOT occurred) & (LSI is VH) $\rightarrow$ (pos is H).
R7.	($vz > 0$) & (vzmag is L) & (dhdg is occurred) & (LSI is VL) $\rightarrow$ (pos is L).
R8.	($vz > 0$) & (vzmag is L) & (dhdg is occurred) & (LSI is L) $\rightarrow$ (pos is M).
R9.	($vz > 0$) & (vzmag is L) & (dhdg is occurred) & (LSI is M) $\rightarrow$ (pos is M).
R10.	($vz > 0$) & (vzmag is L) & (dhdg is occurred) & (LSI is H) $\rightarrow$ (pos is H).
R11.	($vz > 0$) & (vzmag is L) & (dhdg is occurred) & (LSI is VH) $\rightarrow$ (pos is H).
R12.	($vz > 0$) & (vzmag is VL) & (dhdg is occurred) & (LSI is VL) $\rightarrow$ (pos is M).
R13.	($vz > 0$) & (vzmag is VL) & (dhdg is occurred) & (LSI is L) $\rightarrow$ (pos is M).
R14.	($vz > 0$) & (vzmag is VL) & (dhdg is occurred) & (LSI is M) $\rightarrow$ (pos is H).
R15.	($vz > 0$) & (vzmag is VL) & (dhdg is occurred) & (LSI is H) $\rightarrow$ (pos is H).
R16.	($vz > 0$) & (vzmag is VL) & (dhdg is occurred) & (LSI is VH) $\rightarrow$ (pos is VH).
R17.	($vz < 0$) & (altitude is NOT VL) & (delivery is NOT unlikely) & (LSI is VL) $\rightarrow$ (pos is M).
R18.	($vz < 0$) & (altitude is NOT VL) & (delivery is NOT unlikely) & (LSI is L) $\rightarrow$ (pos is H).
R19.	($vz < 0$) & (altitude is NOT VL) & (delivery is NOT unlikely) & (LSI is M) $\rightarrow$ (pos is H).
R20.	($vz < 0$) & (altitude is NOT VL) & (delivery is NOT unlikely) & (LSI is H) $\rightarrow$ (pos is VH).
R21.	($vz < 0$) & (altitude is NOT VL) & (delivery is NOT unlikely) & (LSI is VH) $\rightarrow$ (pos is VH).
R22.	($vz < 0$) & (delivery is unlikely) & (LSI is VL) $\rightarrow$ (pos is L).
R23.	($vz < 0$) & (delivery is unlikely) & (LSI is L) $\rightarrow$ (pos is M).
R24.	($vz < 0$) & (delivery is unlikely) & (LSI is M) $\rightarrow$ (pos is M).
R25.	($vz < 0$) & (delivery is unlikely) & (LSI is H) $\rightarrow$ (pos is H).
R26.	($vz < 0$) & (delivery is unlikely) & (LSI is VH) $\rightarrow$ (pos is H).

$\mu_F(\cdot)$, defined by

$$z_{COA} = \frac{\int_Z \mu_F(z) z\, dz}{\int_Z \mu_F(z) dz}.$$

4. CONFORMANCE MONITORING

In conventional air traffic control and air traffic management operations, the controller creates a visualization of the current and future state dynamics of all aircraft under his control. For each individual aircraft, the controller determines if its observed behavior conforms to the expected or planned path [30, 35]. Unintentional deviations can result from noise in the surveillance systems, atmospheric effects and dynamics of the aircraft navigation systems. Such deviations can be used as threshold values in the definition of a "conformance region." An observed flight profile that lies within the region would be considered conforming, while one that lies beyond the region would be considered non-conforming. In the latter case, knowledge of the conformance status provides a basis for the air traffic controller to implement rectifying measures for the aircraft concerned.

In [31], an analysis framework was developed for the purpose of investigating issues pertaining to conformance monitoring in ATC/ATM. The conformance monitoring task was put forward as a fault detection problem. Fault detection and isolation techniques were used to determine if observable aircraft states were consistent with behavior that was normal (that is, conforming) or abnormal (that is, non-conforming). In other words, non-conforming behavior of an aircraft was regarded as a "fault" to be detected in the ATC/ATM system. The proposed framework comprised the following components:

- conformance basis— basis from which expected state behaviors of an aircraft are generated and against which observed behaviors of the subject aircraft are compared;
- actual system representation— key elements that execute instructions that form the communicated conformance basis;
- conformance monitoring model— generates expected state behaviors against which observed state behaviors are to be compared (requires appropriate level of fidelity to carry out effective conformance monitoring);
- conformance monitoring functions— determine at any time if observed state behaviors are consistent with expected state behaviors that are output by the conformance monitoring model.

The framework was implemented for several conformance monitoring tasks in air traffic control [32–34].

Enhancement and/or improvement of techniques for conformance monitoring is of much interest because of its importance in proper operation of ATC/ATM systems. In addition, there is much awareness of potential hazards to the air transport system posed by non-conforming aircraft that deviate from expected traffic patterns.

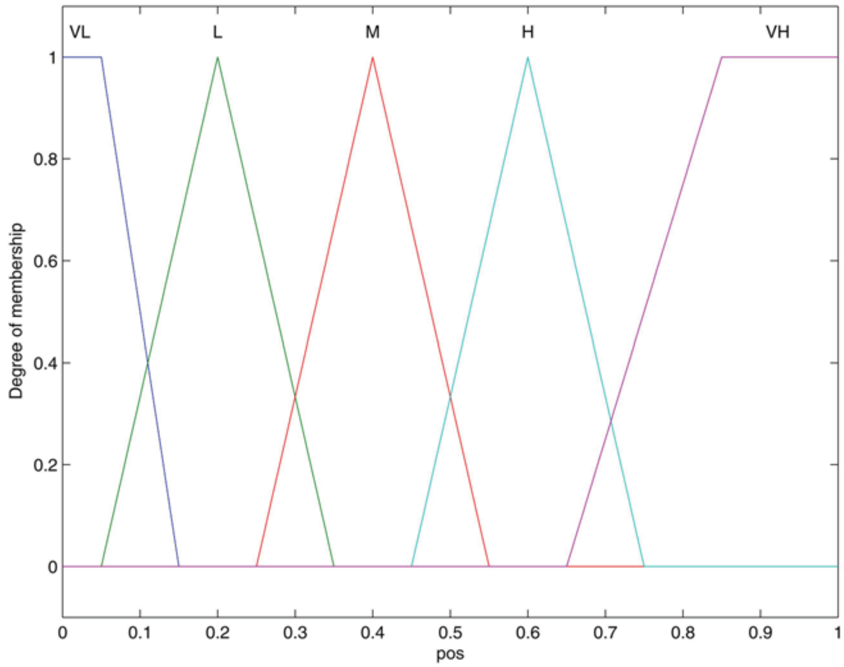


Fig. 11. Membership functions of "pos."

In order to maintain the safety, security and efficiency of ATC/ATM systems, timely detection of non-conforming behavior in aircraft is essential. Our objective in this application is to use a fuzzy inference approach to determine if a tracked aircraft is navigating within conformance limits.

4.1. Process and Techniques

The proposed procedure (a slight modification of Procedure 1 in Section 3.2) for inferring the possibility of non-conformance in the behavior of a tracked aircraft is stated below.

Procedure 2

1. For an aircraft under surveillance, record its state information (sensor measurement data) through observation.
2. Apply the IMM algorithm [22, 24] to update the track state estimates.
3. Fuzzy inference process
   a. Input relevant parameters of the filtered flight trajectory to a Mamdani-type fuzzy inference system generated using the MATLAB Fuzzy Logic Toolbox [38, 39].
   b. Output produced by the FIS is the inferred possibility of non-conformance in the behavior of the tracked aircraft.

The system diagram for the proposed approach is identical to that shown in Fig. 3, omitting the consideration of environmental context. Fig. 4 shows the fuzzy inference process, with input and output variables replaced by those described in Section 4.1.1.

4.1.1. Fuzzy Inference Process

Firstly, fuzzification of the input variables is as described in Section 3.2.1. The input variables considered in the current application are obtained from kinematic parameters of the filtered flight trajectory. Each of the input variables, with respect to the tracked aircraft, is defined below.

The first variable is the deviation of the estimated position from the planned position (measured in feet, abbreviated *dp*). The second variable is the deviation of the estimated velocity from the planned velocity (measured in feet per second, abbreviated *dv*). The third variable is the deviation of the estimated heading from the planned heading (measured in radians, abbreviated *dh*). Figs. 12 to 14 show the membership functions for the three input variables. The symbols and their corresponding linguistic values for membership functions are shown in Table I (where applicable).

Next, rule evaluation (application of the fuzzy operator in the antecedent, followed by implication from the antecedent to the consequent) is carried out as stated in Sections 3.2.2 and 3.2.3. The rules used in the current application are listed in Table III. They are based on predetermined threshold values for state deviations in the definition of a "conformance region." Fig. 15 shows the membership functions for the output variable (inferred possibility of non-conformance in the behavior of the tracked aircraft, abbreviated *pnc*).

As mentioned in Sections 3.2.4 and 3.2.5, the output fuzzy sets (assigned by the consequents) of each rule are aggregated to form a single output fuzzy set. Defuzzification of this final output fuzzy set yields the output of the fuzzy inference system, which is a single crisp/non-fuzzy number.

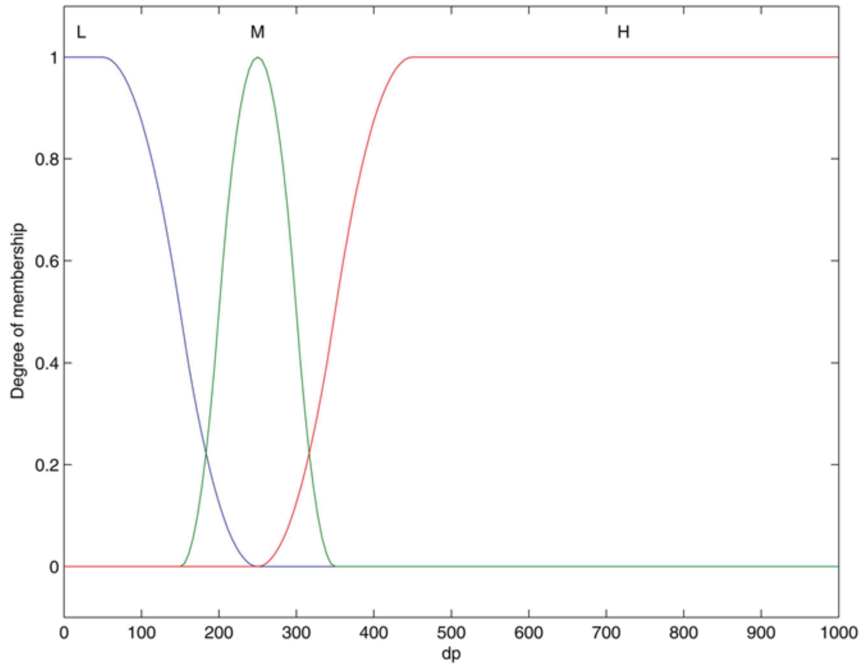


Fig. 12. Membership functions of "dp."

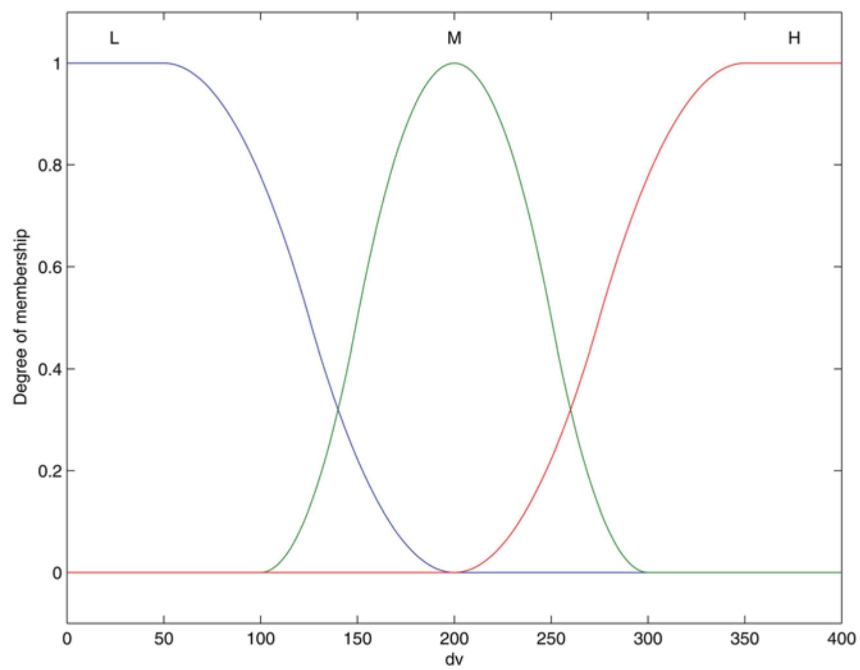


Fig. 13. Membership functions of "dv."

5. SIMULATION TESTS AND RESULTS

We carry out simulation tests to verify the plausibility of the proposed approach. The state estimation component of the method is as follows. Consider a three-dimensional kinetic model described by the discrete-time dynamic system

$$X_{k+1} = f(X_k, w_k) \tag{1}$$

and the measurement/observation equation

$$Z_{k+1} = h(X_{k+1}, v_{k+1}). \tag{2}$$

At time step k, the state vector is $X_k = [x_k, y_k, z_k, \dot{x}_k, \dot{y}_k, \dot{z}_k]^T$. The process noise vector w_k is assumed to be white Gaussian with covariance matrix Q. The measurement vector is Z_k and the measurement noise vector v_k is assumed to be white Gaussian with covariance matrix R. Scalar matrices are used for Q and R. The sampling interval is $T = 1$ (second).

The IMM algorithm used in this section comprises a constant velocity model and two coordinated turn models (one left-turn and one right-turn). The transi-

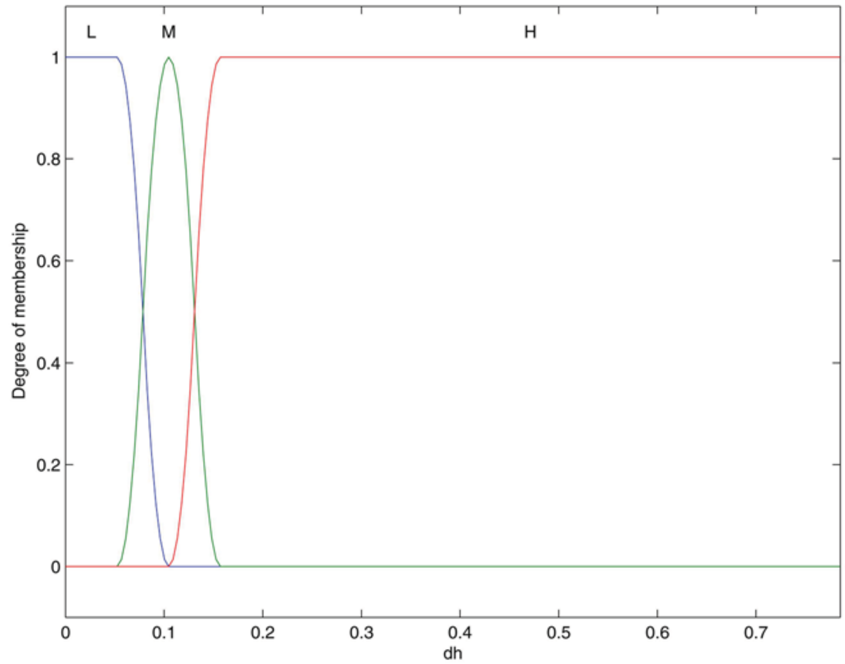


Fig. 14. Membership functions of "dh."

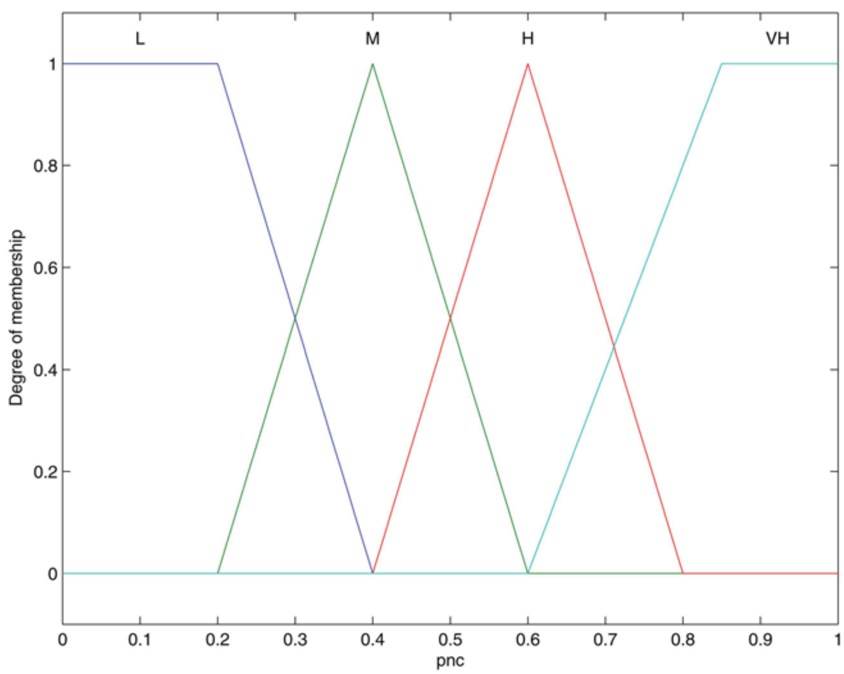


Fig. 15. Membership functions of "pnc."

tion probability matrix and the initial mode probability are

$$\begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \qquad \text{and} \qquad [0.9 \quad 0.05 \quad 0.05]$$

respectively. The choices made for the transition probability matrix values [2, 29] are based on the following reasons. The frequency of mode switches for a tracked target is expected to be low, compared to that of it staying in the same mode (that is, remaining in the same type of motion). The probability of a switch from the current mode to another is expected to be the same for each of the remaining modes. The expected sojourn time of the system in the constant velocity mode is likely to be higher than in the other modes. In addition, the two coordinated turn models only differ in their turning directions, so the transition probabilities for them are set in the same way.

TABLE III
Rules for Fuzzy Inference System
(Conformance Monitoring)

R1.	(dp is L) & (dv is L) & (dh is L) → (pnc is L)
R2.	(dp is L) & (dv is L) & (dh is M) → (pnc is M)
R3.	(dp is L) & (dv is L) & (dh is H) → (pnc is M)
R4.	(dp is L) & (dv is M) & (dh is L) → (pnc is M)
R5.	(dp is L) & (dv is M) & (dh is M) → (pnc is M)
R6.	(dp is L) & (dv is M) & (dh is H) → (pnc is H)
R7.	(dp is L) & (dv is H) & (dh is L) → (pnc is M)
R8.	(dp is L) & (dv is H) & (dh is M) → (pnc is H)
R9.	(dp is L) & (dv is H) & (dh is H) → (pnc is VH)
R10.	(dp is M) & (dv is L) & (dh is L) → (pnc is M)
R11.	(dp is M) & (dv is L) & (dh is M) → (pnc is M)
R12.	(dp is M) & (dv is L) & (dh is H) → (pnc is H)
R13.	(dp is M) & (dv is M) & (dh is L) → (pnc is M)
R14.	(dp is M) & (dv is M) & (dh is M) → (pnc is H)
R15.	(dp is M) & (dv is M) & (dh is H) → (pnc is VH)
R16.	(dp is M) & (dv is H) & (dh is L) → (pnc is M)
R17.	(dp is M) & (dv is H) & (dh is M) → (pnc is H)
R18.	(dp is M) & (dv is H) & (dh is H) → (pnc is VH)
R19.	(dp is H) & (dv is L) & (dh is L) → (pnc is M)
R20.	(dp is H) & (dv is L) & (dh is M) → (pnc is H)
R21.	(dp is H) & (dv is L) & (dh is H) → (pnc is VH)
R22.	(dp is H) & (dv is M) & (dh is L) → (pnc is M)
R23.	(dp is H) & (dv is M) & (dh is M) → (pnc is H)
R24.	(dp is H) & (dv is M) & (dh is H) → (pnc is VH)
R25.	(dp is H) & (dv is H) & (dh is L) → (pnc is H)
R26.	(dp is H) & (dv is H) & (dh is M) → (pnc is VH)
R27.	(dp is H) & (dv is H) & (dh is H) → (pnc is VH)

5.1. Weapon Delivery by Attack Aircraft

We use the simulation results for the following test examples to evaluate the effectiveness of the proposed method.

EXAMPLE 1. Aircraft in surveillance region of low to high LSI.

We use computation formulas in [40] to determine pop-up delivery parameters. Simulation is carried out on 100 different flight trajectories which are generated using various pop-up delivery parameter values.

For each test, as described in Procedure 1 (see Section 3.2), the IMM algorithm is applied to update the state vectors obtained from each flight trajectory. In the filter used, the discrete-time dynamic system of each model is of the form represented by Equations 1 and 2. Next, for each state estimate, determine the environmental context and the corresponding location sensitivity index.

Let A denote the xy-plane (horizontal plane) portion of the entire surveillance region, with the navigation convention (azimuth = 0 along the positive y-axis). Consider the partition

$$A = \bigcup_{i=1}^{2} \bigcup_{j=1}^{8} A_{ij}$$

where
A_{1j} is the jth octant with $x^2 + y^2 < B_2^2$, $j = 1,\ldots,8$,
A_{2j} is the jth octant with $B_2^2 \leq x^2 + y^2 < B_1^2$,

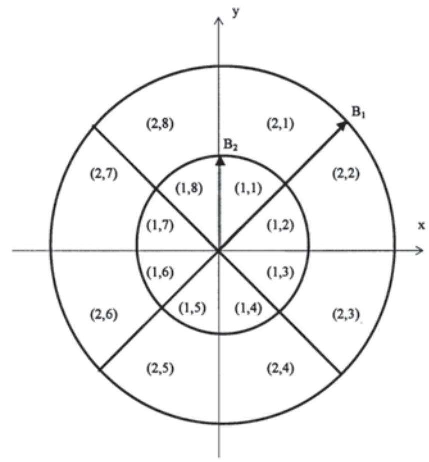


Fig. 16. Partition of surveillance region (xy-plane).

$j = 1,\ldots,8$, and bounds B_1 and B_2 are given positive constants.

The environmental contexts of the partition subsets of A are predetermined and can vary. Let M be a given matrix corresponding to the partition of A, where the LSI for each partition subset A_{ij} is $M(i,j)$, $i = 1,2$, $j = 1,\ldots,8$. Fig. 16 shows the layout for A, with each partition subset denoted according to its subscript by (i,j), $i = 1,2$, $j = 1,\ldots,8$. For each state estimate X_k of the flight trajectory obtained from the filtering process, use the position components x_k and y_k to identify the partition subset, $A_{i(k),j(k)}$, that X_k is in and the corresponding LSI, $M(i(k),j(k))$. The relevant parameters of the flight trajectory obtained from the filtering process and the LSI obtained for the track state estimates are input to a Mamdani-type fuzzy inference system generated using the MATLAB Fuzzy Logic Toolbox [38, 39]. The output produced by the fuzzy inference system is the inferred possibility of the tracked aircraft carrying out a weapon delivery. In this application, we propose to classify a tracked aircraft as having adversarial intent when the fuzzy inference system output exceeds 0.85.

Fig. 17 shows typical results obtained at different phases of the filtered flight trajectory (lower graph), in a scenario where the tracked aircraft travels from regions of low to high sensitivity (and LSI). In the upper graph, the solid curve shows the FIS output values (denoted by P henceforth, in this and subsequent test examples) obtained with only the flight profile considered during simulation. The dash-dot curve shows the FIS output values (denoted by P' henceforth, in this and subsequent test examples) obtained via simulation with both the flight profile and the environmental context of the tracked aircraft considered. Table IV shows P and P' corresponding to the five specific points (defined in Section 3) on the filtered flight trajectory.

It can be observed that P increases as time passes during the tracking process. The surge in P at scan 19 is triggered by motion that is characterized/interpreted by the FIS as the onset of transition from the climb-

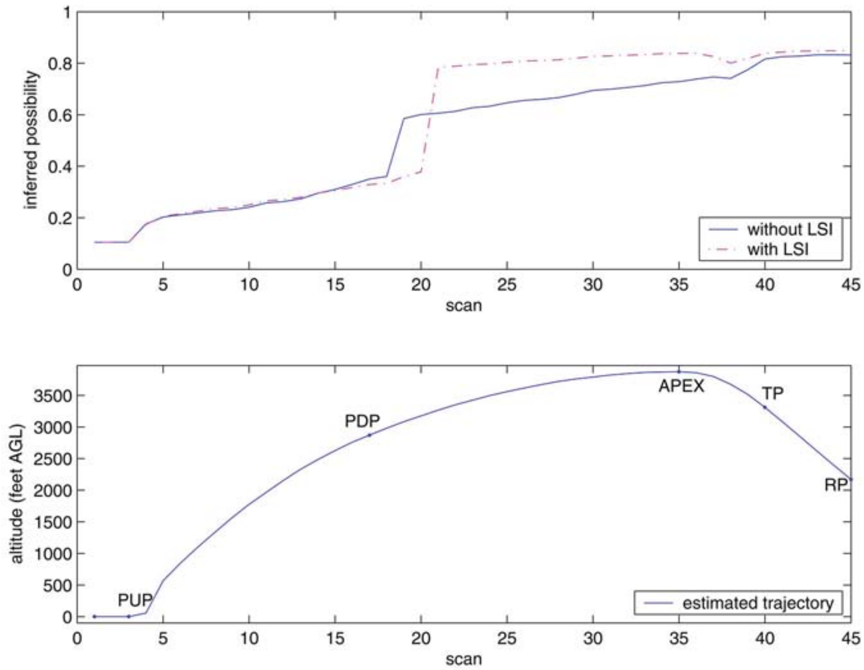


Fig. 17. Example 1—Fuzzy inference system output.

TABLE IV
Example 1—Fuzzy Inference System Output
(to 3 decimal places)

Position on Flight Profile	PUP	PDP	Apex	TP	RP
Without LSI	0.105	0.350	0.728	0.816	0.832
With LSI	0.105	0.329	0.838	0.839	0.848

ing to the diving portion of a pop-up delivery. Thus, the FIS returns a significant increase in P, for warning purposes. P attains its peak around (and beyond) the apex. P remains high in the later part of the tracking process. This observation provides verification for the feasibility of our proposed approach for adversarial intent inference, based on the assumption that the aircraft is approaching its weapon release point.

In regions of low (respectively, high) sensitivity, low (respectively, high) corresponding LSI brings about $P' < P$ (respectively, $P' > P$). In the latter situation, the higher P' is likely to be useful in raising military defenders' alert against a potential adversary.

It appears from the simulation results that a tracked aircraft is very likely to carry out a weapon delivery when P (or P') exceeds 0.85. It is probably appropriate for military defenders to raise the level of vigilance when P (or P') exceeds 0.7. This would allow them to have more time to devise and take pre-emptive action against the potential adversary. Fig. 17 shows that P' exceeds 0.7 earlier than P. This provides justification that taking into consideration the environmental context of the tracked aircraft is useful for improving the efficiency of our approach for adversarial intent inference.

EXAMPLE 2. Aircraft in surveillance region of low LSI.

This example is analogous to Example 1, with the entire surveillance region being of low LSI. Typical simulation results obtained are shown in Fig. 18.

The shapes of the plotted curves are similar to the corresponding ones in Fig. 17. During the early stages of tracking, P and P' are low and almost identical. As in Example 1, there is a surge in P at scan 21, which is triggered by motion that is interpreted by the FIS as the onset of transition from climbing to diving portion of a pop-up delivery. Towards the later part of the tracking process, P exceeds 0.7, which is reasonably high. On the other hand, $P' < P$ and remains below 0.6, which is moderate. In addition, P does not exceed 0.75, which is below the proposed threshold value of 0.85 for classifying an aircraft as having adversarial intent.

Compared to Example 1, there appears to be less critical need/urgency in taking action against the tracked aircraft. This is due to the low sensitivity in the surveillance region, which leads to relatively lower P' values when corresponding P values become high. However, it would probably be advisable for the defenders to maintain their vigilance against such an aircraft, whose flight profile closely resembles that of a pop-up delivery.

EXAMPLE 3. Aircraft cruising at high altitude.

We consider an aircraft that cruises at high altitude throughout the approach. Two possible scenarios are described as follows.

EXAMPLE 3a. Aircraft cruising in surveillance region of low to high LSI.

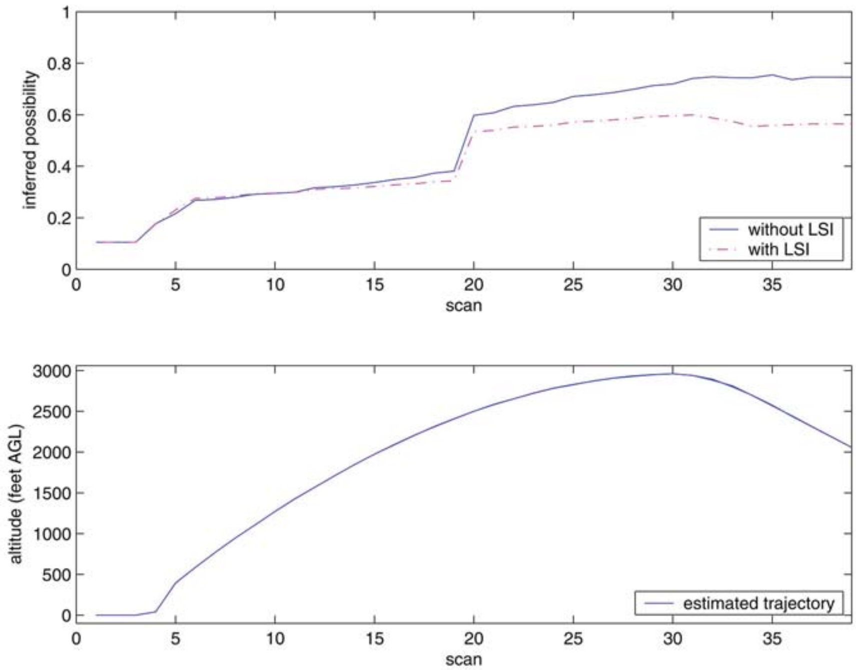


Fig. 18. Example 2—Fuzzy inference system output.

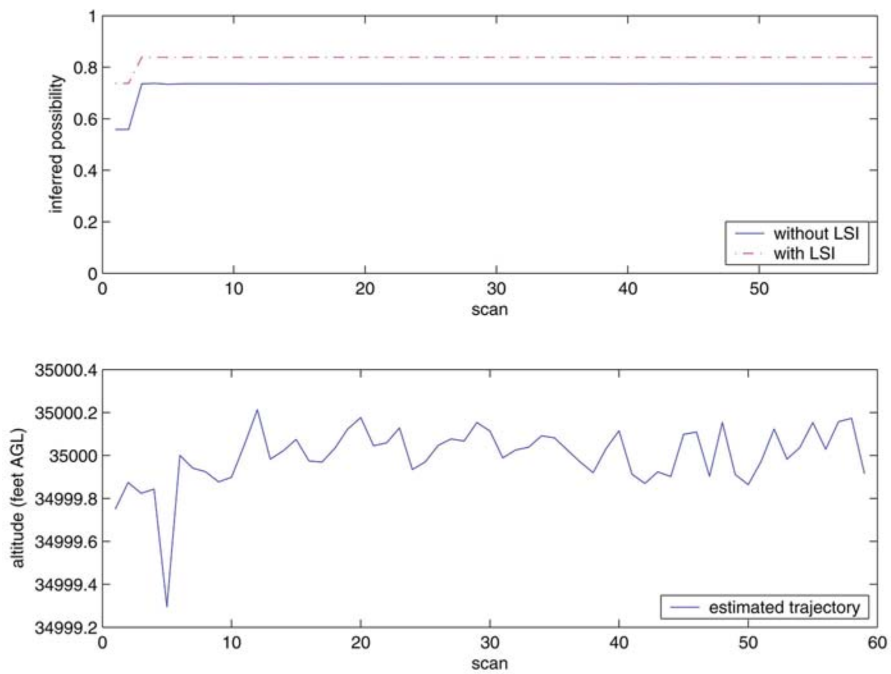


Fig. 19. Example 3a—Fuzzy inference system output.

It can be observed from Fig. 19 that a relatively high value of $P > 0.7$ is reached during tracking. However, there is no further flight motion that indicates an impending attack, which would have caused an increase in P. In this situation, $P' > P$, with $P' \in (0.8, 0.85)$ attained. In view of the high values for P and P', it is very likely for the defenders to be on high alert against possible attack by the aircraft.

EXAMPLE 3b. Aircraft cruising in surveillance region of low LSI.

This example is analogous to Example 3a, with the entire surveillance region being of low LSI. It is apparent from Fig. 20 that the values of P obtained are almost identical to those obtained in Example 3a. Due to the low LSI of the surveillance region, P' remains at a lower level of about 0.5 throughout the approach. It appears from the simulation results that there is no immediate need to raise the defenders' alert against the aircraft.

EXAMPLE 4. Aircraft unlikely to launch an attack.

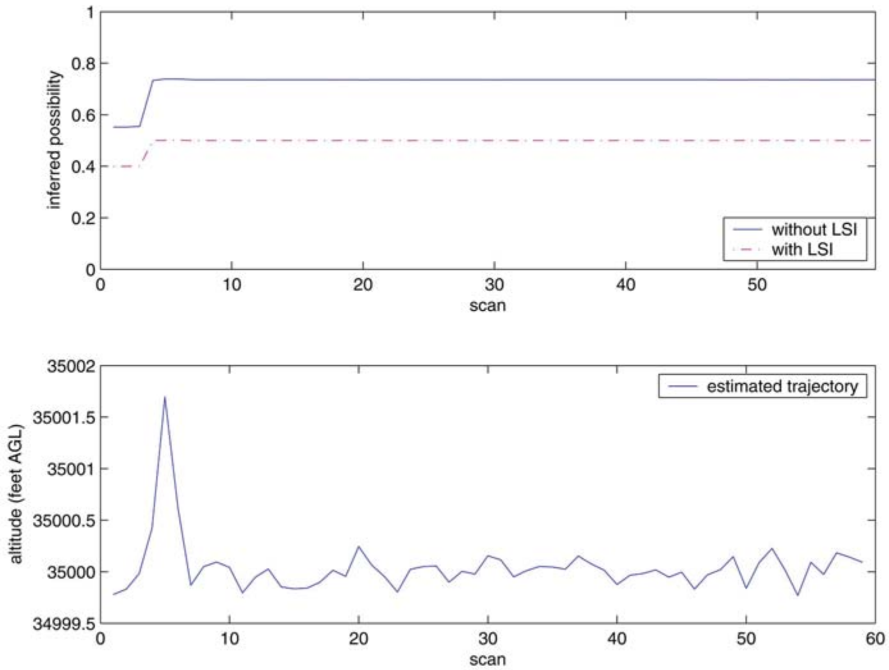


Fig. 20. Example 3b—Fuzzy inference system output.

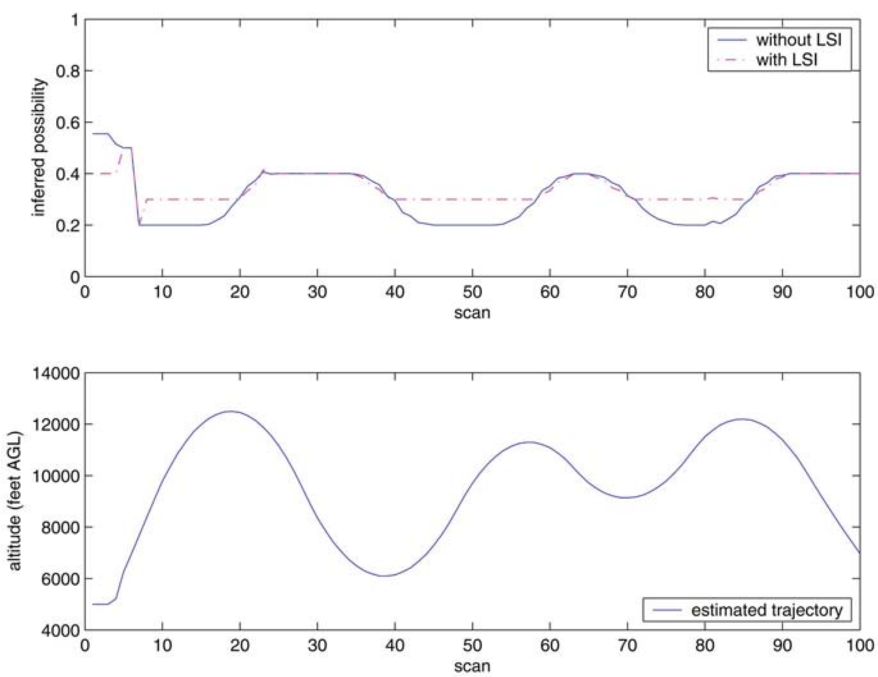


Fig. 21. Example 4—Fuzzy inference system output.

Fig. 21 shows an instance of results obtained for the simulated flight trajectory of an aircraft which is unlikely to carry out a weapon delivery, such as one that is performing aerobatics. It can be seen that P, as well as P', is always below the proposed threshold value of 0.85 for classifying an aircraft as having adversarial intent.

5.2. Conformance Monitoring

Consider the planned flight trajectory shown in Fig. 22. Simulation tests are carried out on 100 flight profiles generated using different combinations of flight parameters (based on existing computation formulas and constraints). For each test, Procedure 2 (see Section 4.1) is carried out to obtain the inferred possibility of non-conformance in the behavior of the tracked aircraft. We categorize aircraft behavior into three types, namely, *conforming*, *non-conforming* and *ambiguous* [31], in our discussion. Fig. 23 depicts typical simulation results obtained.

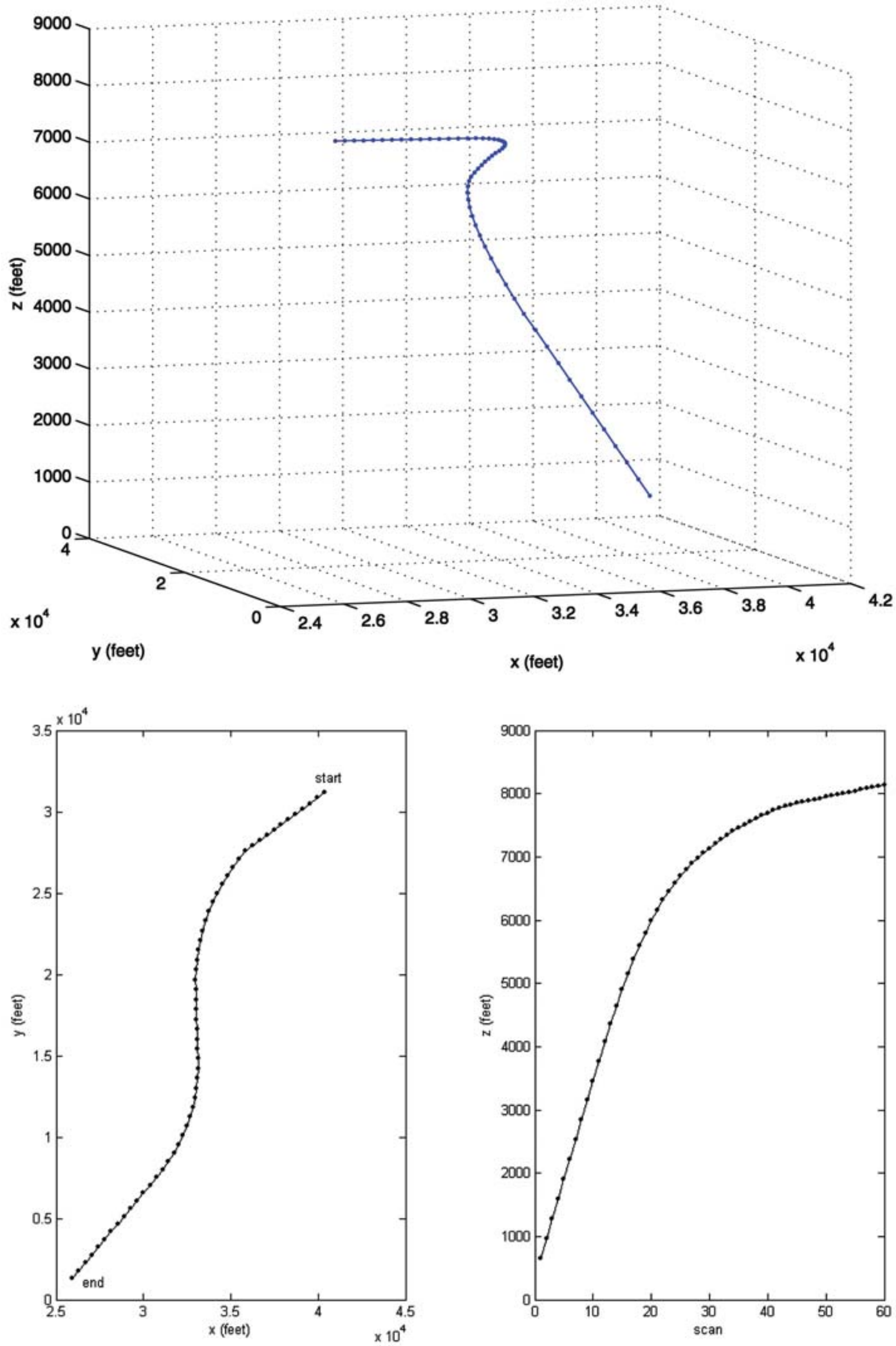


Fig. 22. Planned flight trajectory.

For the conforming case, FIS output values (denoted by P'' henceforth) remain consistently moderate throughout the tracking process. The corresponding deviations from planed states (namely, position, velocity and heading) are relatively small. For the non-conforming case, P'' rises rapidly after an initial period of low to moderate values during tracking. The surge in P'' is due to significant increases in state deviations. The third type of aircraft behavior is considered ambiguous due to indefiniteness in the behavioral traits represented by P''. In this case, there exist instances when P'' increases to become sufficiently large to indicate non-conformance, where corresponding state deviations manifest aberrant behavior in aircraft maneuver. However, P'' subsequently decreases to the extent that conformance is signified, where corresponding state deviations provide evidence of a shift towards the right direction of travel.

It appears from the simulation results that aircraft behavior can be deemed non-conforming when $P'' > 0.85$. It is suggested that alert against non-conformance should be raised when $P'' > 0.7$. This would enable ATC/ATM system controllers to provide the pilot with early warning against navigating beyond safety limits.

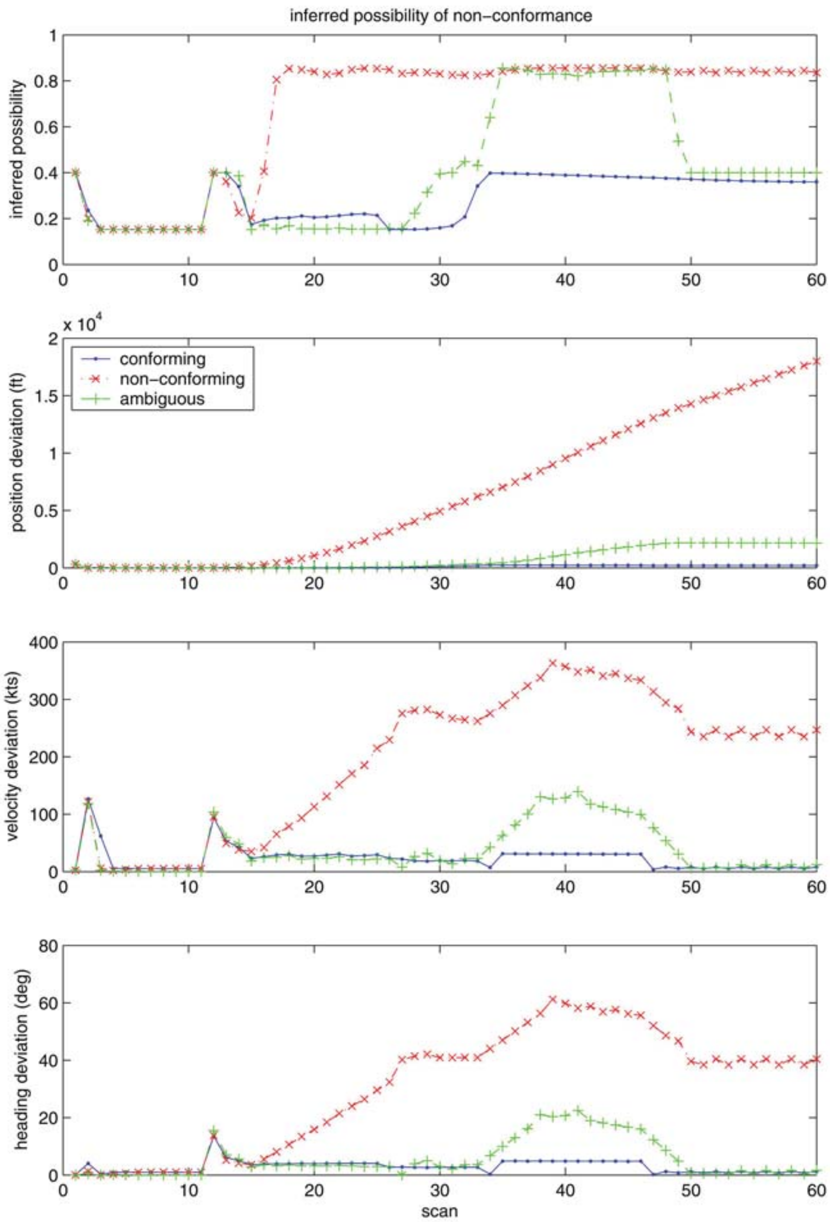


Fig. 23. Fuzzy inference system output (conformance monitoring).

Consequently, the pilot would likely be able to execute necessary maneuvers to steer back towards the planned trajectory with less delay.

6. APPROACH BY MORE THAN ONE AIRCRAFT

Our proposed method deals with intent inference for a single aircraft. The problem on handling an approach by multiple aircraft in military surveillance and air traffic control/management would be more complex and would require much additional consideration. Some issues associated with this problem are discussed below.

6.1. Flight Formation

The flight approach can be in individual form or in a formation. Some examples of flight formations employed by tactical combat aircraft are briefly described in this section [40].

6.1.1. Two-ship Formation

In a line abreast formation, the position of the wingman[2] relative to the flight leader is 0° to 20° aft, 4000 to 12000 feet spacing with altitude separation. A vertical stack of 2000 to 6000 feet is used, when applicable, to minimize the chance of simultaneous detection by an opponent.

For a fighting wing formation, the wingman is given a maneuvering cone from 30° to 70° aft of line abreast and lateral spacing between 500 and 3000 feet. This

[2]Wingman: in a formation of aircraft, the pilot who flies behind and to the side of the leader.

formation is employed when maximum maneuvering potential is desired.

6.1.2. Four-ship Formation

The four-ship formation is employed under the control of one flight leader. It is employed as a single entity as long as it is not forced to separate into a lead element (flight leader and his wingman) and a second/trailing element (second leader and his wingman).

In a box formation, the two-ship elements use basic line abreast maneuvering and principles concerning lookout responsibilities. Depending on terrain and weather, the trailing element takes 1.5 to 3 nautical miles separation from the lead element. The spacing serves the purpose of maximizing separation to avoid easy visual detection of the entire flight formation. maneuvers are initiated by the element leaders in this formation.

For a fluid four formation, the element leaders maintain line abreast formation, while their wingmen assume fighting wing. The flight leader is at the front of the formation, with his wingman to his rear left. The second leader is to the rear right of the flight leader, while his wingman assume fighting wing. The assembly of four of these formations forms a squadron formation.

In a spread four formation, the element leaders maintain the same spacing as for the fluid four formation. The wingmen position themselves 0° to 30° to the rear of their respective element leaders at 6000 to 9000 feet spread. The increase in lateral spacing for wingmen facilitates maneuvering. The elements need not always be line abreast. There may be instances when are briefly in trail. Spread formation makes it difficult to visually acquire the entire flight formation.

A three-ship contingency formation can be considered as an alternative for a four-ship formation mission in some occasions. It is obtained from the four-ship formation concerned by having an appropriate flight member fall out from the original formation.

6.1.3. Echelon Formation

The flight members are arranged diagonally in an echelon formation. Each member is positioned to the rear right, or to the rear left, of the member ahead. These two types of formations are known as a right echelon and a left echelon respectively.

6.2. Multiple Target Tracking and Identity Management

The problem of dealing with approach by more than one aircraft requires the employment of multiple target tracking techniques [4–6, 13, 14, 26, 28] for the state estimation component of our proposed intent inference method. For each tracked aircraft, information based on the estimated kinematic states need to be taken into consideration for processing by a fuzzy inference system, in order to derive the pilot intent.

As mentioned before, the amount of computational load/time is a critical factor for the two intent inference problems discussed here. Hence, it is desirable to select multiple target tracking algorithms with modest time complexities.

Another point of concern is the detection and identification of the targets under surveillance. It may be difficult to distinguish the targets from one another during tracking when there is close proximity and/or interaction among them, such as in the case of a tactical aircraft formation.

To address the aforementioned issues, the multiple-target tracking and identity management (MTIM) algorithm developed in [14] could be considered. The MTIM algorithm is constituted of the following components:

- data association—uses a computationally efficient algorithm based on the joint probabilistic data association algorithm [24], in which measurement data is associated with targets via the use of target kinematic information (position and velocity);
- tracking/hybrid state estimation—uses residual-mean IMM algorithm based on multiple aircraft dynamics models; and
- identity management—uses an algorithm with the ability to keep track of target identities via the use of local attribute information about them (either explicitly available from sensors or inferred from a technique based on the multiple hypothesis tracking algorithm [24]).

The applicability of the MTIM algorithm for incorporation into the intent inference method proposed in this paper could be investigated as part of our future research.

7. SUMMARY

In this paper, we have presented an approach for intent inference, which concerns the use of available knowledge on the preceding activities of a target of interest to predict its future action. The approach is based on the analysis of aircraft flight profiles. The method is implemented for two applications.

Firstly, it has been shown that it is possible to infer the intent of an attack aircraft, particularly on its weapon delivery. The proposed approach is extended to consider the environmental context of the tracked aircraft when executing the inference process. Simulation is carried out on four test examples with different scenarios to evaluate the performance of the method. The results verify the feasibility of the method and its ability to provide timely inference. It is also justifiable to consider the environmental context, which is useful in raising military defenders' level of vigilance early against potential adversaries, hence allowing more time to prepare for pre-emptive action.

In the second application, experimental results show that the proposed solution has much potential in being a useful tool for conformance monitoring in ATC/ATM. It can be used to assist ATC/ATM system controllers in determining whether aircraft are deviating from or adhering to designated courses of travel. As a result, corrective/remedial actions can be taken once deviant behavior is detected.

Our proposed intent inference method has only considered an approach by a single aircraft. We briefly discuss the extension of the proposed method to deal with an approach by multiple aircraft, such as that by a flight formation. Some of the main issues concerned include multiple target tracking and management of the target identities. These topics are of interest in our future research.

APPENDIX. FUZZY LOGIC

Generally, vagueness and imprecision exist in data/information concerning real-world problems. Fuzzy logic [15, 38], an extension of Boolean logic, was developed to deal with uncertainties associated with problems from practical applications.

In *classical set theory*, a set has a crisp (sharp and clear) boundary and it completely includes or excludes an arbitrarily given element. On the other hand, in *fuzzy set theory*, boundaries between sets of values need not be distinctly defined. A fuzzy set expresses the degree to which an element belongs to a set, where an element can have gradual transition in status from "belongs to a set" to "does not belong to a set."

Let X be a space of objects and x be an arbitrary element of X. For a *classical set* C, $C \subseteq X$, define a *characteristic function* $f : X \mapsto \{0,1\}$ by

$$f(x) = \begin{cases} 0, & x \notin C, \\ 1, & x \in C. \end{cases}$$

Then C can be represented by a set of ordered pairs,

$$C' = \{(x, f(x)) \mid x \in X\}. \tag{3}$$

DEFINITION 1 *Fuzzy sets and membership functions*.
Let X be a space of objects which are generically denoted by x. A *fuzzy set* F in X is defined as a set of ordered pairs

$$F = \{(x, \mu_F(x)) \mid x \in X\} \tag{4}$$

where $\mu_F : X \mapsto Y$ is known as the *membership function* for F. The membership function maps each element x of the *input space* (or *universe of discourse*) X to a *degree of membership* (also known as *membership value* or *membership grade*) $\mu_F(x)$ in the *output space* (or *membership space*) Y. For each $x \in X$, $\mu_F(x) \in [0,1]$.

REMARK The definition of a fuzzy set is an extension of the definition of a classical set. In Definition 1, if $Y = \{0,1\}$, then F is reduced to a classical set and $\mu_F(\cdot)$ is the characteristic function of F.

Fuzzy logic is a superset of standard Boolean logic. There exist fuzzy logical operations for fuzzy sets that correspond to Boolean logical operations for classical sets. In the case when membership function values are restricted to the set $\{0,1\}$, fuzzy logical operations and Boolean logical operations are equivalent.

DEFINITION 2 *Fuzzy complement*.
A *fuzzy complement* operator is a continuous function $N : [0,1] \to [0,1]$ that meets the basic axiomatic requirements:

$$\begin{aligned} &N(0) = 1 \text{ and } N(1) = 0 \quad \text{(boundary)} \\ &N(a) \geq N(b) \text{ if } a \leq b \quad \text{(monotonicity).} \end{aligned} \tag{5}$$

An optional requirement imposes *involution* on a fuzzy complement:

$$N(N(a)) = a \quad \text{(involution)} \tag{6}$$

which guarantees that the double complement of a fuzzy set is still the set itself.

The *complement* of a fuzzy set F is the fuzzy set $\bar{F}$ (or $\neg F$, NOT F), whose membership function is related to that of F by

$$\mu_{\bar{F}}(x) = N(\mu_F(x)) \tag{7}$$

with the fuzzy complement operator commonly defined by $N(a) = 1 - a$.

DEFINITION 3 *T-norm*.
A *T-norm* operator is a binary function $T : [0,1] \times [0,1] \to [0,1]$ that satisfies:

$$\begin{aligned} &T(0,0) = 0, \quad T(a,1) = T(1,a) = a \quad \text{(boundary)} \\ &T(a,b) \leq T(c,d) \text{ if } a \leq c \text{ and } b \leq d \quad \text{(monotonicity)} \\ &T(a,b) = T(b,a) \quad \text{(commutativity)} \\ &T(a,T(b,c)) = T(T(a,b),c) \quad \text{(associativity).} \end{aligned} \tag{8}$$

DEFINITION 4 *T-conorm (or S-norm)*.
A *T-conorm* (or *S-norm*) operator is a binary function $S : [0,1] \times [0,1] \to [0,1]$ satisfying:

$$\begin{aligned} &S(1,1) = 1, \quad S(0,a) = S(a,0) = a \quad \text{(boundary)} \\ &S(a,b) \leq S(c,d) \text{ if } a \leq c \text{ and } b \leq d \quad \text{(monotonicity)} \\ &S(a,b) = S(b,a) \quad \text{(commutativity)} \\ &S(a,S(b,c)) = S(S(a,b),c) \quad \text{(associativity).} \end{aligned} \tag{9}$$

DEFINITION 5 *Fuzzy intersection (conjunction)*.
The *intersection* of two fuzzy sets F_1 and F_2 is a fuzzy set F, written as $F = F_1 \cap F_2$ or $F = F_1$ AND F_2. F is specified in general by a T-norm operator $T : [0,1] \times [0,1] \to [0,1]$, which aggregates the membership values of F_1 and F_2 as

$$\mu_F(x) = T(\mu_{F_1}(x), \mu_{F_2}(x)). \tag{10}$$

A frequently used T-norm operator is defined by $T(a,b) = \min(a,b)$, the *minimum* of $\{a,b\}$ (also denoted by $a \wedge b$).

DEFINITION 6 *Fuzzy union (disjunction).*
The *union* of two fuzzy sets F_1 and F_2 is a fuzzy set F, written as $F = F_1 \cup F_2$ or $F = F_1$ OR F_2. F is specified in general by a T-conorm (or S-norm) operator $S : [0,1] \times [0,1] \rightarrow [0,1]$, which aggregates the membership values of F_1 and F_2 as

$$\mu_F(x) = S(\mu_{F_1}(x), \mu_{F_2}(x)). \qquad (11)$$

A frequently used S-norm operator is defined by $S(a,b) = \max(a,b)$, the *maximum* of $\{a,b\}$ (also denoted by $a \vee b$).

For an input vector $x \in X$, a fuzzy inference process utilizes a set of fuzzy rules to interpret the values of x and assign appropriate values to an output vector $y \in Y$. Each rule is of the form "if S_1 then S_2," or equivalently, "$S_1 \rightarrow S_2$." The if-part of the rule "S_1" is called the *antecedent*, while the then-part of the rule "S_2" is called the *consequent*. Each rule outputs a fuzzy set. Aggregation of the output fuzzy sets for the rules yields a single output fuzzy set. Defuzzification is carried out on the resultant set to obtain the final desired conclusion, in the form of a single number.

REFERENCES

[1] A. Aligawesa and K. L. Fetzer
Intent inference and conformance monitoring in air traffic control.
Project Report, Flight Dynamics and Control/Hybrid Systems Laboratory, School of Aeronautics and Astronautics, Purdue University, Apr. 2008.

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

[3] B. Bell, E. Santos, Jr. and S. M. Brown
Making adversary decision modeling tractable with intent inference and information fusion. In *Proceedings of the 11th Conference on Computer-Generated Forces and Behavior Representation*, Orlando, FL, May 7–9, 2002, 535–542.

[4] H. A. P. Blom and E. A. Bloem
Joint particle filtering of multiple maneuvering targets from unassociated measurements.
Journal of Advances in Information Fusion, **1**, 1 (July 2006), 15–34.

[5] H. A. P. Blom and E. A. Bloem
Bayesian tracking of two possibly unresolved maneuvering targets.
IEEE Transactions on Aerospace and Electronic Systems, **43**, 2 (Apr. 2007), 612–627.

[6] B. Chen and J. K. Tugnait
Tracking of multiple maneuvering targets in clutter using IMM/JPDA filtering and fixed-lag smoothing.
Automatica, **37**, 2 (Feb. 2001), 239–249.

[7] S.-B. Cho
Incorporating soft computing techniques into a probabilistic intrusion detection system.
IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, **32**, 2 (May 2002), 154–160.

[8] J. P. Coyne
Bombology.
Air Force Magazine, **73**, 6 (June 1990), 64–69.

[9] P. H. Foo, G. W. Ng, K. H. Ng and R. Yang
Application of intent inference for surveillance and conformance monitoring to aid human cognition.
In *Proceedings of the 10th International Conference on Information Fusion*, Québec, Canada, July 9–12, 2007, paper 1309.

[10] J. Franke, S. M. Brown, B. Bell and H. Mendenhall
Enhancing teamwork through team-level intent inference.
In *Proceedings of the International Conference on Artificial Intelligence*, Las Vegas, NV, June 26–29, 2000.

[11] D. L. Hall and S. A. H. McMullen
Mathematical Techniques in Multisensor Data Fusion (2nd ed).
Boston, MA: Artech House, 2004.

[12] L. D. Hudson, B. S. Ware, K. B. Laskey and S. M. Mahoney
An application of Bayesian networks to antiterrorism risk management for military planners.
Technical Report C4I-05-01, Center of Excellence in Command, Control, Communications, Computing and Intelligence (C4I Center), George Mason University, 2001.

[13] C. Hue, J.-P. Le Cadre and P. Pérez
Sequential Monte Carlo methods for multiple target tracking and data fusion.
IEEE Transactions on Signal Processing, **50**, 2 (Feb. 2002), 309–325.

[14] I. Hwang, H. Balakrishnan, K. Roy and C. Tomlin
Multiple-target tracking and identity management algorithm with application to aircraft tracking.
Journal of Guidance, Control and Dynamics, **30**, 3 (May–June 2007), 641–653.

[15] J.-S. R. Jang, C.-T. Sun and E. Mizutani
Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence.
Upper Saddle River, NJ: Prentice-Hall, 1997.

[16] P. Jarvis, T. F. Lunt and K. L. Myers
Identifying terrorist activity with AI plan recognition technology.
In *Proceedings of the 19th National Conference on Artificial Intelligence, 16th Conference on Innovative Applications of Artificial Intelligence*, San Jose, CA, July 25–29, 2004, 858–863.

[17] L. A. Klein
Sensor and Data Fusion: A Tool for Information Assessment and Decision Making.
Bellingham, WA: SPIE Press, 2004.

[18] A. Kott and W. M. McEneaney (eds.)
Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind.
Boca Raton, FA: Taylor & Francis, 2006.

[19] J. Krozel and D. Andrisani, II
Intent inference with strategic path prediction.
Journal of Guidance, Control, and Dynamics, **29**, 2 (Mar.–Apr. 2006), 225–236.

[20] J. Krozel, T. Mueller and D. Schleicher
Intent inference, confidence assessment, and hazard prioritization status report.
Technical Report TR 99175.30-02, Seagull Technology, Inc., Mar. 2000.

[21] C. G. Looney and L. R. Liang
Cognitive situation and threat assessments of ground battlespaces.
Information Fusion, **4**, 4 (Dec. 2003), 297–308.

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

[23] M. Negnevitsky
Artificial Intelligence: A Guide to Intelligent Systems (2nd ed).
Harlow, England and New York: Addison-Wesley, 2005.

[24] G. W. Ng
Intelligent Systems—Fusion, Tracking and Control.
Baldock, Hertfordshire, England: Research Studies Press Ltd. and Philadelphia, PA: Institute of Physics Publishing, 2003.

[25] G. W. Ng, K. H. Ng, R. Yang and P. H. Foo
Intent inference for attack aircraft through fusion. In *Proceedings of SPIE, Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2006*, Orlando, FL, Apr. 19–20, 2006, 6242-06.

[26] S. Oh and S. Sastry
An efficient algorithm for tracking multiple maneuvering targets.
In *Proceedings of the IEEE Conference on Decision and Control, and the European Control Conference*, Seville, Spain, Dec. 12–15, 2005, 4010–4015.

[27] M. T. Plehn
Control warfare: Inside the OODA loop.
Master's thesis, School of Advanced Airpower Studies, Air University, Maxwell Air Force Base, 2000.

[28] S. Puranik and J. K. Tugnait
Tracking of multiple maneuvering targets using multiscan JPDA and IMM filtering. *IEEE Transactions on Aerospace and Electronic Systems*, **43**, 1 (Jan. 2007), 23–35.

[29] K. V. Ramachandra
Kalman Filtering Techniques for Radar Tracking.
New York: Marcel Dekker, 1999.

[30] T. G. Reynolds and R. J. Hansman
Analysis of aircraft separation minima using a surveillance state vector approach. In *Proceedings of the FAA/Eurocontrol International Air Traffic Management R&D Seminar*, Naples, Italy, June 13–16, 2000.

[31] T. G. Reynolds and R. J. Hansman
Conformance monitoring approaches in current and future air traffic control environments.
In *Proceedings of the IEEE/AIAA 21st Digital Avionics Systems Conference*, Irvine, CA, Oct. 27–31, 2002, 2, 7.C.1-1–7.C.1-12.

[32] T. G. Reynolds and R. J. Hansman
Analyzing conformance monitoring in air traffic control using fault detection approaches & operational data.
In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Austin, TX, Aug. 11–14, 2003, paper AIAA-2003-5574.

[33] T. G. Reynolds and R. J. Hansman
Investigating conformance monitoring issues in air traffic control using fault detection techniques.
Journal of Aircraft, **42**, 5 (Sept.–Oct. 2005), 1307–1317.

[34] T. G. Reynolds, R. J. Hansman and H. Li
Investigating fundamental issues in lateral conformance monitoring using a fault detection approach.
In *Proceedings of the 5th Eurocontrol/FAA ATM R&D Seminar*, Budapest, Hungary, June 23–27, 2003.

[35] T. G. Reynolds, J. M. Histon, H. J. Davison and R. J. Hansman
Structure, intent & conformance monitoring in ATC.
In *Proceedings of the Workshop on Advanced Technologies and Their Impact on Air Traffic Management in the 21st Century*, Capri, Italy, Sept. 22–26, 2002.

[36] D. D. Salvucci
Inferring driver intent: a case study in lane-change detection.
In *Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting*, New Orleans, LA, Sept. 20–24, 2004, 2228–2231.

[37] E. Santos, Jr.
A cognitive architecture for adversary intent inferencing: Knowledge structure and computation.
In *Proceedings of SPIE, 17th Annual International Symposium on Aerospace/Defense Sensing and Controls*, Orlando, FL, Apr. 21–25, 2003, 5091, 182–193.

[38] The MathWorks, Inc.
Fuzzy Logic Toolbox User's Guide Version 2, 2002.

[39] The MathWorks, Inc.
Fuzzy Logic Toolbox Version 2.1.2 (Release 13), 2002.

[40] United States Air Force Air Combat Command
Multi-Command Handbook 11-F16 Vol. 5, May 1996.

[41] J. L. Yepes, I. Hwang and M. Rotea
New algorithms for aircraft intent inference and trajectory prediction.
Journal of Guidance, Control, and Dynamics, **30**, 2 (Mar.–Apr. 2007), 370–382.

[42] Y. Zhao, C. Haissig and M. J. Hoffman
Analysis of pilot intent parameters in air traffic management.
In *Proceedings of the American Control Conference*, Philadelphia, PA, June 24–26, 1998, 3, 1789–1792.

Pek Hui Foo received her B.Sc. (Hons.), M.Sc. and Ph.D. degrees from the National University of Singapore in 1999, 2001, and 2009, respectively.

She is currently a Senior Member of Technical Staff at DSO National Laboratories. Her research is on applications of data and information fusion.

Gee Wah Ng received his M.Sc. and Ph.D. from University of Manchester Institute of Science and Technology, United Kingdom.

He is currently a Principal Member of Technical Staff and Programme Director (Information Exploitation) of Information Division at DSO National Laboratories. He has delivered many projects in the decision support areas and has authored two books. He is active in international conferences in the areas of information fusion and intelligent systems. His research interests in data and information fusion include target tracking, computational intelligence, machine learning, self-tuning and sensor networks.

Khin Hua Ng received his B.Sc. degree (with 1st class honours) in computational science and mathematics from the National University of Singapore in 1998, and his M.Sc. degree (with distinction) in advanced computing from Imperial College London in 2004.

He is currently a Senior Member of Technical Staff at DSO National Laboratories. His research interests include sensor management, target recognition, intent inference, probabilistic reasoning, and cognitive system.

Rong Yang received her B.E. degree in information and control from Xi'an Jiao Tong University China in 1986, and M.Sc. degree in electrical engineering from the National University of Singapore in 2000.

She is currently a Senior Member of Technical Staff at DSO National Laboratories. Her research interests include target tracking and data fusion.

On Fusion of Multiple Objectives for UAV Search & Track Path Optimization

VESSELIN P. JILKOV

X. RONG LI

This paper addresses the problem of designing a fused scalar objective function for autonomous surveillance—target search and tracking (S&T)—by unmanned aerial vehicles (UAVs). A typical S&T mission includes multiple, most often inherently conflicting, objectives such as detection, survival, and tracking. A common approach to coping with this issue is to optimize a fused scalar objective—a convex combination (weighted sum) of the individual objectives. In practice, determining the fusion weights of a multiobjective combination is, more or less, a guesswork whose success is highly dependent on the designer's assessment and intuition. An optimal (trade-off) point in the performance space is hard to come up with by varying the weights of the individual objectives. In this paper the problem of designing optimal fusion weights is treated more systematically in a rigorous multiobjective optimization (MOO) framework. The approach is based on finding a set of optimal points (Pareto front) in the performance space and solving the inverse problem—determine the fusion weights corresponding to a chosen optimal performance point. The implementation is done through the known normal boundary intersection (NBI) numerical method for computing the Pareto front. The use of the proposed methodology is illustrated by several case studies of typical S&T scenarios.

Manuscript received October 20, 2007; revised July 13, 2008; released for publication December 18, 2008.

Refereeing of this contribution was handled by Jean-Pierre LeCadre.

A preliminary version of this paper was presented at the 10th International Conference on Information Fusion, July 7–12, 2007, Québec City, Canada.

Research supported in part by ARO through grant W911NF-08-1-0409 and NAVO through Contract N62306-09-P-3S01.

Authors' address: Department of Electrical Engineering, University of New Orleans, New Orleans, LA 70148, E-mail: (vjilkov@uno.edu, xli@uno.edu)

1557-6418/09/$17.00

1. INTRODUCTION

Due to the significant advancement of the *unmanned vehicle* (UV) technologies in recent years, a great deal of research effort has been devoted to the problem of path optimization (planning and dynamic replanning) of a single or multiple UVs in uncertain and possibly hostile environments. While various UV mission scenarios have been considered in the literature, this paper is focused on UAV surveillance missions which typically include search (detection and localization) of new targets and possibly tracking of detected targets. The techniques considered, however, can be easily applied to other types of missions as well.

Most of the literature on autonomous UAV surveillance deals with search oriented systems, e.g., [8], [7], [4], [15], [14]. Multiple-UAV tracking has been addressed in [9], [12], and tracking combined with detection has been dealt with in [10], [11]. In all of its variations an S&T mission includes multiple objectives, often conflicting to each other. At a high level these objectives can be grouped into several different types including, but not limited to, target *detection*, target *tracking* (classification, recognition), UAV *survivability*, UAV *cooperation*, UAV *efficiency*, and possibly others [7]. Quantifying various objectives and defining a fused scalar *mission objective function* to be optimized during a mission is a crucial issue in the design of S&T systems. Commonly, search-only systems use mission objective functions made up of, most often probability-based, gain/loss functions—e.g., cumulative detection probability, survival probability, etc. [8], [7], [4], [15], [14]. The tracking oriented systems of [9], [12] use information gain based mission objectives, in terms of the Fisher information matrix (FIM) of the tracking filters, and [10], [11] further include the detection objective measured also in terms of FIM. This makes it possible to use standard estimation fusion techniques [1] to fuse the detection and estimation objectives into a scalar objective. However, expressing all objectives through FIMs is difficult to extend to more complicated practical scenarios, e.g., to include efficiency (UAV flight regime cost) or other objectives.

Achieving the mission goal is inherently a multiobjective optimization (MOO) problem and in this paper the problem of designing a mission objective function is treated as such—within the framework of the MOO methodology. There are two issues associated with the MOO formulation. First, due to the conflict among the individual objectives the solution in general is not unique. There is a set of optimal points (referred to as Pareto front) such that, loosely speaking, each optimal point corresponds to a certain trade-off among the values of the objective functions. A decision has to be made as to which Pareto optimal point provides the "best trade-off" among all the alternatives. The second issue is implementational—solving an MOO problem by the known computational methods is usually associated with solving a great number of single nonlinear optimization problems and thus is not feasible in "real-time" for an S&T mission.

A natural approach that circumvents these issues is to optimize a *weighted sum* (WS)—convex combination—of the individual objectives as a fused scalar mission objective. Any unique solution of a WS optimization problem is Pareto optimal, and for each Pareto optimal point there exists a set of weights such that solving the WS problem yields this point if the MOO problem is convex. These properties as well as its simplicity is what makes the WS objective attractive for online implementation in S&T problems. It should be noted that WS objectives have been used in a number of algorithms for cooperative UAV target search systems [8], [7], [15]. In the sequel we also assume that the mission objective function used for online optimization, referred to as a fused mission objective, is a WS of the individual mission objectives. Our focus is on the problem of determining the fusion weights in an optimal manner when a WS fused mission objective is designed.

Implementing a WS as a fused mission objective for online UAV flight path optimization presumes knowledge of the fusion weights and its effectiveness depends heavily on these weights. In practice, their specification is done *a priori*, based on subjective considerations about, e.g, the importance of the individual objective functions. It is more or less a guesswork whose success is highly dependent on the designer's assessment and intuition, and other uncertain factors. For example, [8], [7], [15] state that priorities to the specific individual objectives can be achieved by "adjusting" the values of the weights. However, to make such an adjustment optimally is a nontrivial task for the designer. The problem is that the choice of the weights based on importance or priorities is not made in the feasible objective space—the real *performance space*. For complicated nonlinear and conflicting objective functions (such as in an S&T mission), a "reasonable choice" of importance weights may lead to a rather unacceptable trade-off (Pareto optimal point) in the performance space. At the same time acceptable trade-offs may be available for other, non-obvious choices of fusion weights. In addition a trade-off point in the performance space is hard to come up with by simply varying the weights of the individual objectives.

We argue that a more systematic and rigorous way for designing the fusion coefficients that achieve the "best trade-off" among the possible alternatives is needed. Our approach is based on finding a set of optimal points (Pareto front) and solving the inverse problem—determine the fusion weights corresponding to a chosen optimal performance point. The implementation is done through the normal boundary intersection (NBI) numerical method of [3] for computing the Pareto front.

The underlying idea for application of our methodology is to design the weights of the fused criterion for path optimization such that an acceptable trade-off is achieved by this criterion when applied online to real-life scenarios. This can be done by a comprehensive trade-off analysis through Monte Carlo simulation of an ensemble of typical S&T mission scenarios with different detection maps, threat models, efficiency functions, etc. Overall, the approach proposed in this paper is intended to facilitate the design of fused mission objective functions through more insightful determination of the fusion coefficients of the individual objectives.

The mathematical formulation of the problem is given in Section 2. Section 3 provides some necessary background information about MOO and describes an algorithmic solution. Results of several case studies, illustrating the use of the proposed methodology, are presented in Section 4. Conclusions are provided in Section 5.

2. PROBLEM FORMULATION

We consider a team of UAVs engaged in searching a given surveillance region for new (undetected) targets and tracking of detected targets in an uncertain, dynamic, and risky environment. A UAV (sensor suite, or just sensor for short) is denoted by s, $s = 1,2,\ldots,N_s$; a (detected) target that is being tracked is denoted by t, $t = 1,2,\ldots,N_t$. The 2D surveillance region is partitioned into N_n cells numbered by $n = 1,2,\ldots,N_n$ and $p_n = (x_n, y_n)$ denotes the center location of the nth cell in Cartesian coordinates. n will also stand for indexing a new (undetected) target at position $p_n = (x_n, y_n)$, i.e., in cell n.

Next we present modeling of several objective functions involved in typical UAV S&T mission scenarios, and then we discuss the multiple objectives of a single UAV.

2.1. Detection

For a sensor s, a detection event is modeled through the *detection probability* $\pi_D^s = \pi_D(p_s, p)$, where $p_s = (x_s, y_s)$ and $p = (x, y)$ denote sensor and target locations, respectively. In general π_D^s is a function of the sensor type and parameters, target type and parameters, sensor-target geometry, environment, etc. Here for simplicity we consider the dependence of π_D^s only on the distance between p_s and p. For example, for a ground moving target indicator (GMTI) sensor a typical *detection function* $\pi_D^s = \pi_D(p_s, p) = \pi_D(\|p_s - p\|)$ with $\|p_s - p\| = \sqrt{(x_s - x)^2 + (y_s - y)^2}$, similar to the one used in [10], is shown in Fig. 1.

The probability of detecting a target t, known to exist, by sensor s is $\pi_D^{s,t} = \pi_D(p_s, p_t) = \pi_D(\|p_s - p_t\|)$. If a new target n exists at a given location $p_n = (x_n, y_n)$ with probability $\pi_E(p_n)$, then the probability of detection by sensor s is $\pi_D^{s,n} = \pi_D(\|p_s - p_n\|)\pi_E(p_n)$. The *target existence*[1] *probabilities* $\{\pi_E(p_n)\}_{n=1}^{N_n}$ are assumed known

[1] More precisely, it should be target *perceivability* [5], which is, however, beyond the scope of this paper.

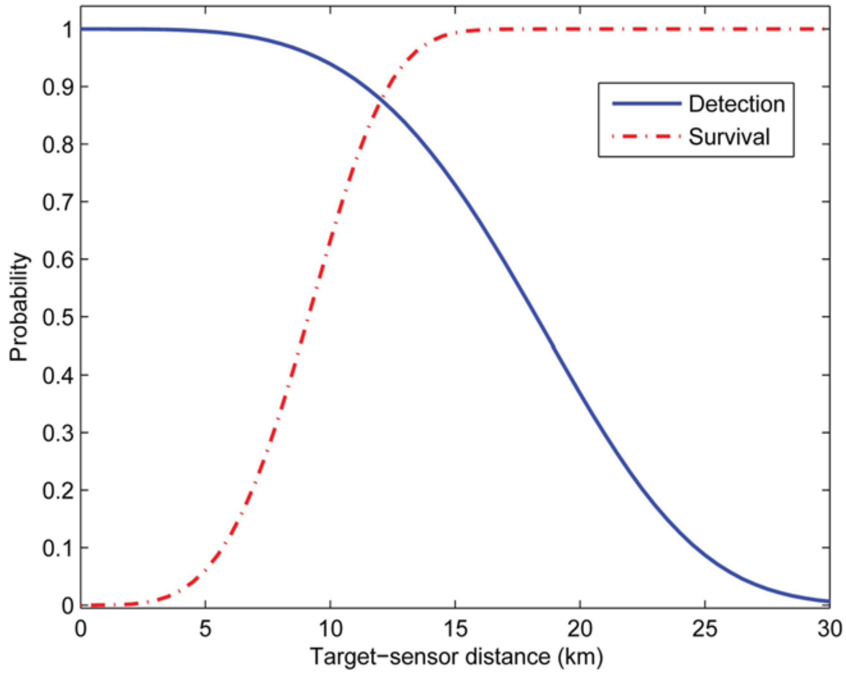


Fig. 1. Detection and survival probabilities.

to the UAVs during the S&T mission via a map of the expected distribution of targets over the surveillance region.

2.2. Survivability

It is assumed that each target (either new or being tracked) poses a threat to a UAV. The event that a sensor s will survive a fire from a given threat θ located at p_θ is modeled through the *survival probability function*[2] $\pi_S(\|p_s - p_\theta\|)$. The probability of surviving the threat from an existing target t is $\pi_S^{s,t} = \pi_S(\|p_s - p_t\|)$. If a threat (new target n) exists at location p_n with probability $\pi_E(p_n)$ then the probability of survival is

$$\pi_S^{s,n} = 1 - (1 - \pi_S(\|p_s - p_n\|))\pi_E(p_n). \tag{1}$$

A survival probability function, assumed in the simulation, is shown in Fig. 1.

2.3. Tracking

For simplicity we assume that a Kalman filter is used for tracking. The tracking objective of a UAV s that tracks a target t can be quantified through the filter *information matrix* (IM) $I = P^{-1}$ where P is the filter covariance matrix.[3] We adopt $\ln|I|$ as a scalar measure[4] of I, and the following approximate relationship for updating the *expected* IM [10]

$$I = \bar{I} + \pi_S \pi_D H' R^{-1} H \tag{2}$$

where $\bar{I}$ is the predicted IM, H is the measurement matrix, and R is the measurement error covariance. Thus the expected *tracking information gain* (TIG) γ_T is measured by

$$\gamma_T = \ln|I| - \ln|\bar{I}| = \ln|\bar{I} + \pi_S \pi_D H' R^{-1} H| - \ln|\bar{I}|. \tag{3}$$

The expected TIG for sensor target pair (s,t) is a function of the distance $\|p_s - p_t\|$, i.e., $\gamma_T^{s,t} = \gamma_T(\|p_s - p_t\|)$ through $\pi_S^{s,t} = \pi_S(\|p_s - p_t\|)$, $\pi_D^{s,t} = \pi_D(\|p_s - p_t\|)$, and $R = R(\|p_s - p_t\|)$ since the observation error depends on the distance. The tracking objective of s is to maximize $\gamma_T(\|p_s - p_t\|)$ with respect to p_s.

2.4. Other Objectives/Constraints

There are a number of other relevant objectives, such as cooperation, engagement, efficiency, whose detailed analysis is not needed for the description of our approach. The reader is referred to, e.g., [7], [15] for a formulation and a more detailed analysis of other objectives. We limit our consideration here to the above three objectives since they are the most significant for an S&T mission but our approach is not limited to these objectives—it allows other objectives to be easily incorporated.

2.5. Fusion of Multiple Objectives

We explain the approach and formulate the problem for a generic single UAV S&T mission scenario since

[2]For simplicity, collision with other UAVs is ignored here but it can be easily modeled in a similar manner.

[3]The superscript indices s, t and the subscript time index k are dropped here to simplify notation.

[4]An alternative scalar measure that can be used is $\mathrm{tr}(I)$.

the mission objectives of a group of UAVs strongly depend on many other factors such as the overall system network architecture (distributed or centralized), cooperation strategy, communication capabilities, whose consideration is beyond the scope of this paper. The approach is, however, directly applicable to such multiple UAV scenarios as well.

Denote by s the sensor under consideration, by $n_1, n_2, \ldots, n_{R_s}$ the cells that can be reached by the sensor in the next time step, and by $t_1, t_2, \ldots, t_{T_s}$ the targets that are being tracked by s. Let

$$\boldsymbol{\pi}_D^s = [\pi_D^{s,n_1}, \ldots, \pi_D^{s,n_{R_s}}]'$$

$$\boldsymbol{\pi}_S^s = [\pi_S^{s,n_1}, \ldots, \pi_S^{s,n_{R_s}}, \pi_S^{s,t_1}, \ldots, \pi_S^{s,t_{T_s}}]'$$

$$\boldsymbol{\gamma}_T^s = [\gamma_T^{s,t_1}, \gamma_T^{s,t_2}, \ldots, \gamma_T^{s,t_{T_s}}]'$$

be the vectors of the detection, survival and tracking objective functions of s.

Given target locations p_{n_i}, $i = 1, \ldots R_s$ and p_{t_j}, $j = 1, \ldots T_s$, the *immediate* (one time-step ahead) goal of s can be rigorously formulated as the following MOO problem

$$\max_{p_s} \begin{bmatrix} \boldsymbol{\pi}_D^s(p_s) \\ \boldsymbol{\pi}_S^s(p_s) \\ \boldsymbol{\gamma}_T^s(p_s) \end{bmatrix} \tag{4}$$

where p_s is the sensor position at the next time.

The dimension of the vector problem (4) can be significantly reduced if the threats are assumed independent. In this case the vector objective $\boldsymbol{\pi}_S^s(p_s)$ can be replaced by the scalar objective

$$\pi_S^s(p_s) = \prod_{i=1}^{R_s} \pi_S^{s,n_i} \prod_{j=1}^{T_s} \pi_S^{s,t_j}.$$

As motivated in Section 1, in order to avoid the problems associated with a complete mathematical solution of (4) for online implementation we assume that the mission objective for online optimization is formulated as the following WS single objective optimization problem

$$\max_{p_s} [\mathbf{w}_D' \boldsymbol{\pi}_D^s(p_s) + \mathbf{w}_S' \boldsymbol{\pi}_S^s(p_s) + \mathbf{w}_T' \boldsymbol{\gamma}_T^s(p_s)] \tag{5}$$

where $\mathbf{w} = [\mathbf{w}_D', \mathbf{w}_S', \mathbf{w}_T']'$ is a fusion weight vector with components $w_i \geq 0$ and $\sum_i w_i = 1$.

We aim at finding numerically (off line) the Pareto front for problem (4) and for each point $(\boldsymbol{\pi}_D^s(p_s^*), \boldsymbol{\pi}_S^s(p_s^*), \boldsymbol{\gamma}_T^s(p_s^*))$ on the front determining the fusion weight vector $\mathbf{w}^*$ such that the solution of (5) is p_s^*, where $\mathbf{w}^*$ is the "best" fused combination of objectives *given* the trade-off point in the performance space $(\boldsymbol{\pi}_D^s(p_s^*), \boldsymbol{\pi}_S^s(p_s^*), \boldsymbol{\gamma}_T^s(p_s^*))$.

3. SOLUTION METHODOLOGY

3.1. MOO Background Concepts

Here we provide brief information about some basic concepts of the MOO needed later. For details the reader is referred to [6], [13].

A multiobjective optimization problem in mathematical notation is posed as follows

$$\text{minimize } \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_M(\mathbf{x}) \end{bmatrix}, \qquad M \geq 2 \tag{6}$$

$$\text{subject to } \mathbf{x} \in C = \{\mathbf{x} : \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, M$, are the objective functions, $\mathbf{x}$ is the *decision variable* vector, C is the *feasible set*, and $\mathbf{h}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ are the *constraint* functions. Usually $\mathbf{f}(\mathbf{x})$, $\mathbf{g}(\mathbf{x})$, and $\mathbf{h}(\mathbf{x})$ are assumed twice continuously differentiable. The image of the feasible set $\mathbf{f}(C) \subseteq \mathbb{R}^M$ is referred to as *feasible objective set*, which is a subset of the objective space $\mathbb{R}^M$.

In general, no single $\mathbf{x}$ exists that minimizes every f_i simultaneously. A common concept of optimality for MOO problems is that of Pareto optimality. A decision vector $\mathbf{x}^* \in C$ is *Pareto optimal* (PO) if there does not exist another decision vector $\mathbf{x} \in C$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ for all $i = 1, \ldots, M$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one index j. An objective vector $\mathbf{y}^* = \mathbf{f}(\mathbf{x}^*)$ is PO if its corresponding decision vector $\mathbf{x}^*$ is PO. Simply put, an objective vector is PO if any attempt to improve a component (individual objective) will deteriorate at least an other component (individual objective). The set of all PO objective vectors is referred to as the Pareto optimal set, or *Pareto front* (PF). The complete mathematical solution is to find the PF.

There are usually infinitely many PO solutions. In practice solving an MOO means finding a PO solution that satisfies the needs and requirements of a *decision maker*. This is usually a person (e.g., a designer or end-user) that supposedly has an insight into the problem, can express preference relations between different solutions, and can select a final, "best," single solution. Such an approach has been used in engineering applications to facilitate solving complex design problems. The power of using the PF stems from the fact that it reveals the entire spectrum of efficient alternatives for a particular practical problem and allows to select the "best" among them.

Perhaps the most natural approach to the MOO problem is that of the *weighted sum* (WS): minimize a convex combination of the individual objectives

$$\text{minimize } \mathbf{w}'\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{M} w_i f_i(\mathbf{x}) \tag{7}$$

$$\text{subject to } \mathbf{x} \in C$$

where $w_i \geq 0$ and $\sum_{i=1}^{M} w_i = 1$. Any unique solution of (7) is PO for the problem (6), and for each PO point of (6) there exists a weighting vector $\mathbf{w}$ such that solving (7) yields this point if the problem (6) is convex [6]. Unfortunately, despite the above properties, finding points on the PF by varying the weighting coefficients $\mathbf{w}$ has been found to suffer serious drawbacks. It has been observed that small changes in $\mathbf{w}$ may cause dramatic changes in the objective vectors and large changes in $\mathbf{w}$ may result in almost unnoticeable changes in the objective vectors. This instability is due to the fact that the WS is not a Lipschitzian function of $\mathbf{w}$ [6]. Clearly, this makes the relation between weights and performance very complicated and *non-intuitive*. Obtaining a good approximation of the PF directly, by uniform sampling of $\mathbf{w}$, may be extremely inefficient since it may lead to very uneven sampling of the PF [2].

An effective method for numerical computation of evenly distributed points on the PF for the MOO problem (6) is the normal boundary intersection (NBI) method of [3]. This method suits very well our "inverse" problem, formulated at the end of Section 3—determine the fusion weights corresponding to a chosen optimal performance point—since it provides a direct link between the NBI computed PF points and their corresponding weights in the problem (7).

3.2. Fusion Weights Determination via NBI

A formal description of the algorithm that we use for determination of the fusion weights through the NBI computed PO points is given below. Its validity follows from Claims 1 and 2, Section 6 of [3].

ALGORITHM

(I) NBI WEIGHTS*: Generate* $\boldsymbol{\beta} = [\beta_1,\ldots,\beta_M]'$ *such that* $\beta_i \geq 0$ *and* $\sum_{i=1}^{M}\beta_i = 1$.

(II) NBI MINIMIZER*: Obtain a point* $\mathbf{x}^* = \mathbf{x}^*_\beta$ *by solving (numerically) the nonlinear optimization problem*

$$\begin{aligned}&\min_{\mathbf{x},t} -t\\ &\text{s.t. } \Phi\boldsymbol{\beta} + t\hat{\mathbf{n}} = \mathbf{f}(\mathbf{x}) - \mathbf{f}^*\\ &\mathbf{x}\in C\end{aligned} \tag{8}$$

determined by computing the following:

1) $\mathbf{x}^*_i = \arg\min_{\mathbf{x}\in C} f_i(\mathbf{x}),\ i = 1,\ldots,M$*—minimizers of the individual objectives of* (6)*;*

2) $\mathbf{f}^* = [f_1(\mathbf{x}^*_1), f_2(\mathbf{x}^*_2),\ldots,f_M(\mathbf{x}^*_M)]'$*—vector of individual minima (utopia point) of* (6)*;*

3) $\Phi = [\mathbf{f}(\mathbf{x}^*_1)-\mathbf{f}^*, \mathbf{f}(\mathbf{x}^*_2)-\mathbf{f}^*,\ldots,\mathbf{f}(\mathbf{x}^*_M)-\mathbf{f}^*]$*—pay-off matrix of* (6)*;*

4) $\hat{\mathbf{n}} = -\Phi[1,1,\ldots,1]'$*—quasi-normal search direction for* (8).

(III) FUSION WEIGHTS*: Obtain* $\mathbf{w}^* = [w^*_1,\ldots,w^*_M]'$ *which corresponds to* $\mathbf{x}^*$ *as*

$$w^*_i = \frac{\lambda_i^{(1)^*}}{\sum_{i=1}^{M}\lambda_i^{(1)^*}}, \qquad i = 1,\ldots,M$$

if all $\lambda_i^{(1)^*}$, $i = 1,\ldots,M$ *have the same signs, where* $\boldsymbol{\lambda}^{(1)^*} = [\lambda_1^{(1)^*},\ldots,\lambda_M^{(1)^*}]'$ *is the vector of the Karush-Kuhn-Tucker (KKT) multipliers for the equality constraint* $\Phi\boldsymbol{\beta} + t\hat{\mathbf{n}} = \mathbf{f}(\mathbf{x}) - \mathbf{f}^*$.

REMARK 1 In our Matlab program implementation of the above algorithm, in Step II, we used the standard function for nonlinear constrained minimization `fmincon` from the Matlab optimization toolbox for minimizing the individual objectives of (6) and solving the problem (8). This function provides the KKT multipliers $\lambda_i^{(1)^*}$, $i = 1,\ldots,M$, needed in Step III, as output parameters.

REMARK 2 As shown in [3], the WS problem (7) with $\mathbf{w} = \mathbf{w}^*$ determined in Step III has the solution $\mathbf{x}^* = \mathbf{x}^*_\beta$ determined in Step II. If $\mathbf{w}^*$ cannot be determined in Step III, i.e., some $\lambda_i^{(1)^*}$ has a sign which is different from the sign of $\sum_{i=1}^{M}\lambda_i^{(1)^*} \neq 0$ then either the NBI computed point $\mathbf{x}^* = \mathbf{x}^*_\beta$ is not PO or $\mathbf{x}^*$ is PO but lies in a nonconvex part of the PF and cannot be obtained by minimizing a WS of the objectives. For convex problems (as most real problems are) such an issue does not exist.

REMARK 3 An even spread of NBI points $\{\mathbf{x}^*_{\beta_\nu}\}_{\nu=1}^N$ will be obtained if the set of points $\{\Phi\boldsymbol{\beta}_\nu\}_{\nu=1}^N$ forms an uniformly-spaced grid on the simplex $\{\Phi\boldsymbol{\beta}\}_{\boldsymbol{\beta}}$. This is due to the fact that, according to (8), the points obtained by the NBI are restricted to lie on a set of parallel vectors (all parallel to the normal $\hat{\mathbf{n}}$) emanating from the uniformly spread points $\{\Phi\boldsymbol{\beta}_\nu\}_{\nu=1}^N$. A simple algorithm to achieve this is to generate the NBI weights $\{\boldsymbol{\beta}_\nu\}_{\nu=1}^N$ uniformly, i.e., each component of $\boldsymbol{\beta}_\nu$ has a value in $[0,1/p,2/p,\ldots,1]$ where $p \geq 2$ is an integer and all components sum up to 1. This yields an uniform grid with a total of $N = \binom{M+p-1}{p}$ points.

4. CASE STUDIES

As formulated in Section 2, a WS single objective optimization problem given by (5) is to be solved online during an S&T mission. The fusion weights $\mathbf{w} = [\mathbf{w}'_D, \mathbf{w}'_S, \mathbf{w}'_T]'$ are designed (determined off-line) based on a comprehensive trade-off analysis such that an acceptable trade-off will be achieved by the WS criterion when applied online to real-life scenarios. As illustrated below, such an analysis can be done through Monte Carlo simulation of an ensemble of typical S&T mission scenarios with different detection maps, threat models, efficiency functions, etc. It includes obtaining a representative set of trade-off points $\{(\boldsymbol{\pi}^s_D, \boldsymbol{\pi}^s_S, \boldsymbol{\gamma}^s_T)\}$ for the problem (4), along with their corresponding weights $\{\mathbf{w}\}$ in the problem (5), and can be done efficiently by means of the NBI-based algorithm of Section 3.2. A decision upon the "best" trade-off point $(\boldsymbol{\pi}^{s*}_D, \boldsymbol{\pi}^{s*}_S, \boldsymbol{\gamma}^{s*}_T)$, made by the designer, gives in turn the "best" fusion weights $\mathbf{w}^*$ to be implemented in the online optimization problem (5).

To illustrate the use of the proposed technique in the trade-off analysis for determination of the "best" fusion weights we present next four case studies of UAV S&T scenarios.

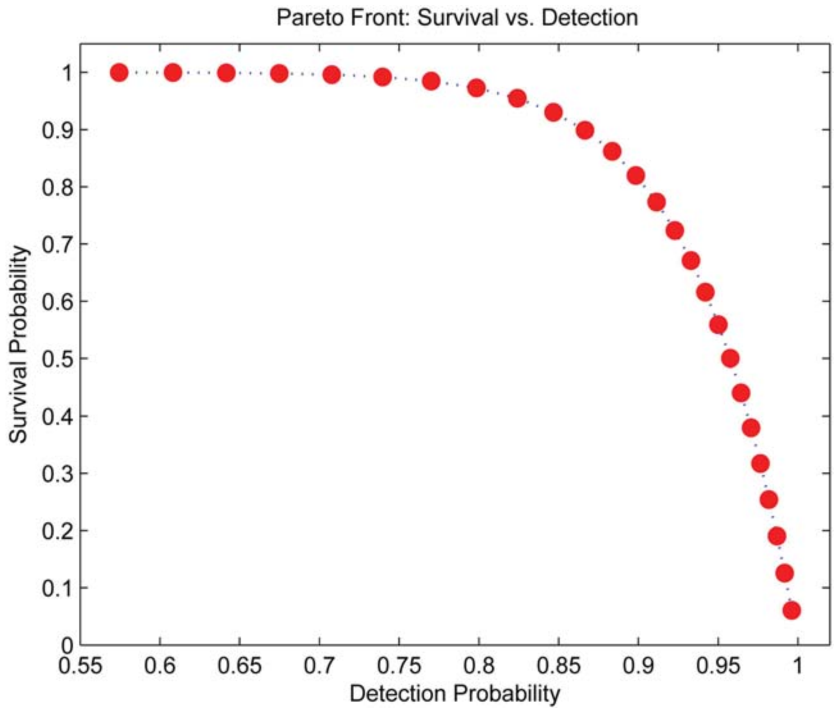


Fig. 2. PF generated by NBI.

4.1. Detection vs. Survivability

1) *Single target search*: For a trade-off analysis between the detection and survivability objectives we consider first a simple scenario of one sensor s and one new target n located at $p_n = (x_n, y_n)$. The UAV s aims at solving the following two-objective optimization problem

$$\max_{p_s} \begin{bmatrix} \pi_D(\|p_s - p_n\|) \\ \pi_S(\|p_s - p_n\|) \end{bmatrix}. \tag{9}$$

By using the algorithm of Section 3.2 we obtain a uniform representation of the Pareto front

$$\pi_D(\|p_s - p_n\|) \text{ vs. } \pi_S(\|p_s - p_n\|)$$

and for each trade-off point $(\pi_D(\|p_s^* - p_n\|), \pi_S(\|p_s^* - p_n\|))$ of the PF we determine the corresponding weights w_D^* and w_S^* such that the solution of the single-objective optimization problem

$$\max_{p_s} [w_D^* \pi_D(\|p_s - p_n\|) + w_S^* \pi_S(\|p_s - p_n\|)] \tag{10}$$

is p_s^*.

The simulated scenario parameters are as follows. The assumed detection and survival functions, shown in Fig. 1, are

$$\pi_D(d) = \exp(-(d/20)^4) \tag{11}$$

$$\pi_S(d) = 1 - \exp(-(d/10)^4) \tag{12}$$

where $d = \|p_s - p_n\|$ is the distance. Without loss of generality it is assumed that $p_n = (0,0)$.

The Pareto front generated by the algorithm of Section 3.2 is shown in Fig. 2. Its computation required solving 24 nonlinear single objective optimization problems of the type (8). For a rough comparison, the direct "bruteforce" WS method for PF determination required a dramatically larger number of problems (5) in order to provide a comparable representation of the PF.

Next, for each optimal point on the PF $(\pi_D(\|p_s^* - p_n\|), \pi_S(\|p_s^* - p_n\|))$ we determined the corresponding fusion weights w_D^* and w_S^* for the equivalent WS single objective optimization problem. The results are given in Table I. It reveals the available trade-offs between the detection and survival probabilities and includes the corresponding weights that yield these trade-offs through maximizing the WS objective. What is left to the user (or designer) is to choose one or more preferable trade-off points and they will be automatically achieved through the corresponding weights. For example, if the selected trade-off performance from Table I is $\pi_D^* = 0.8665$ and $\pi_S^* = 0.899$ (line 12) then the fusion weights $w_D^* = 0.651$ and $w_S^* = 0.349$ are to be used in (10). It should be also noted that Table I allows to design a set of WSs corresponding to different tactical situations and thus give the UAV a capability to operate in different modes depending on the situation by simply switching the weights of the WS objective function.

2) *Multiple target search*: Next we present a trade-off analysis of the detection and survival objectives of a sensor s in a search scenario with two targets n_1 and n_2 known to exist at $p_{n_i} = (x_{n_i}, y_{n_i})$, $i = 1,2$.

For simplicity it is assumed that the threats are independent, and thus the joint survival probability

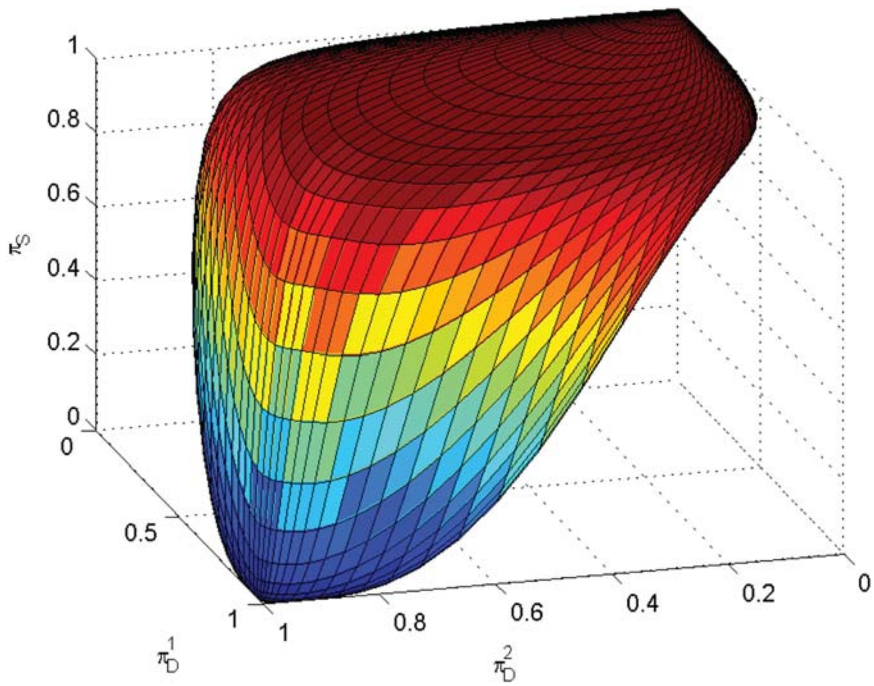


Fig. 3. Feasible objective set.

TABLE I
Trade-Off Points & Fusion Weights

π_D	π_S	w_D	w_S
0.9916	0.1256	0.9338	0.0662
0.9818	0.2541	0.9240	0.0760
0.9764	0.3172	0.9180	0.0820
0.9643	0.4406	0.9027	0.0973
0.9575	0.5006	0.8930	0.1070
0.9419	0.6161	0.8670	0.1330
0.9329	0.6711	0.8494	0.1506
0.9228	0.7237	0.8273	0.1727
0.9114	0.7735	0.7991	0.2009
0.8984	0.8198	0.7624	0.2376
0.8836	0.8619	0.7143	0.2857
0.8665	0.8990	0.6510	0.3490
0.8468	0.9301	0.5690	0.4310
0.8241	0.9548	0.4677	0.5323
0.7984	0.9727	0.3534	0.6466
0.7701	0.9847	0.2412	0.7588
0.7396	0.9920	0.1478	0.8522
0.7077	0.9960	0.0822	0.9178
0.6749	0.9981	0.0421	0.9579
0.6417	0.9992	0.0202	0.9798
0.6081	0.9996	0.0091	0.9909

is

$$\pi_S^s = \pi_S^{s,n_1}\pi_S^{s,n_2} = \pi_S(\|p_s - p_{n_1}\|)\pi_S(\|p_s - p_{n_2}\|)$$

where the function $\pi_S(d)$ is given by (12).

The UAV s aims at solving the following three-objective optimization problem

$$\max_{p_s} \begin{bmatrix} \pi_D(\|p_s - p_{n_1}\|) \\ \pi_D(\|p_s - p_{n_2}\|) \\ \pi_S^s(\|p_s - p_{n_1}\|, \|p_s - p_{n_2}\|) \end{bmatrix} \tag{13}$$

where $\pi_D(d)$ is given by (11).

TABLE II
Trade-Off Points & Fusion Weights

$\pi_D^{n_1}$	$\pi_D^{n_2}$	π_S^s	$w_D^{n_1}$	$w_D^{n_2}$	w_S^s
0.98	0.98	0.04	0.4240	0.4240	0.1520
0.95	0.95	0.27	0.4466	0.4466	0.1068
0.92	0.92	0.50	0.4383	0.4383	0.1234
0.88	0.88	0.72	0.4112	0.4112	0.1776
0.83	0.83	0.89	0.3298	0.3298	0.3404
0.73	0.73	0.98	0.1130	0.1130	0.7740
0.59	0.59	0.99	0.0059	0.0059	0.9882
0.44	0.44	1.00	0.0001	0.0001	0.9998

Note that without the above threat independence assumption π_S^{s,n_1} and π_S^{s,n_2} should be considered as individual objectives (as in the general MOO problem formulation (4)), which would lead to a four dimensional problem.

The parameters of the simulation are the same as in the previous scenario. In addition, the second target n_2 is located at $p_n = (10,0)$.

Fig. 3 shows the *feasible objective region* for detection and survival, and Fig. 4 shows the obtained Pareto front.

Table II gives the fusion weights $w_D^{n_1*}$, $w_D^{n_2*}$ and w_S^{s*} of the WS objective function

$$w_D^{n_1*}\pi_D(\|p_s - p_{n_1}\|) + w_D^{n_2*}\pi_D(\|p_s - p_{n_2}\|) + w_S^*\pi_S^s(\|p_s - p_{n_1}\|, \|p_s - p_{n_2}\|)$$

corresponding to the obtained Pareto optimal points $(\pi_D(\|p_s^* - p_{n_1}\|), \pi_D(\|p_s^* - p_{n_2}\|), \pi_S^s(\|p_s^* - p_{n_1}\|, \|p_s^* - p_{n_2}\|))$.

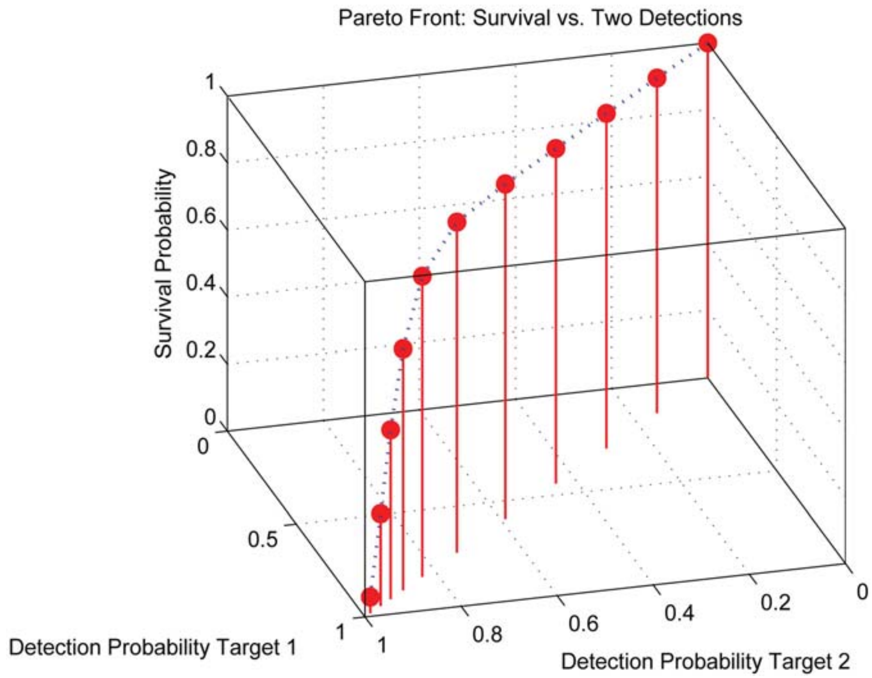


Fig. 4. PF generated by NBI.

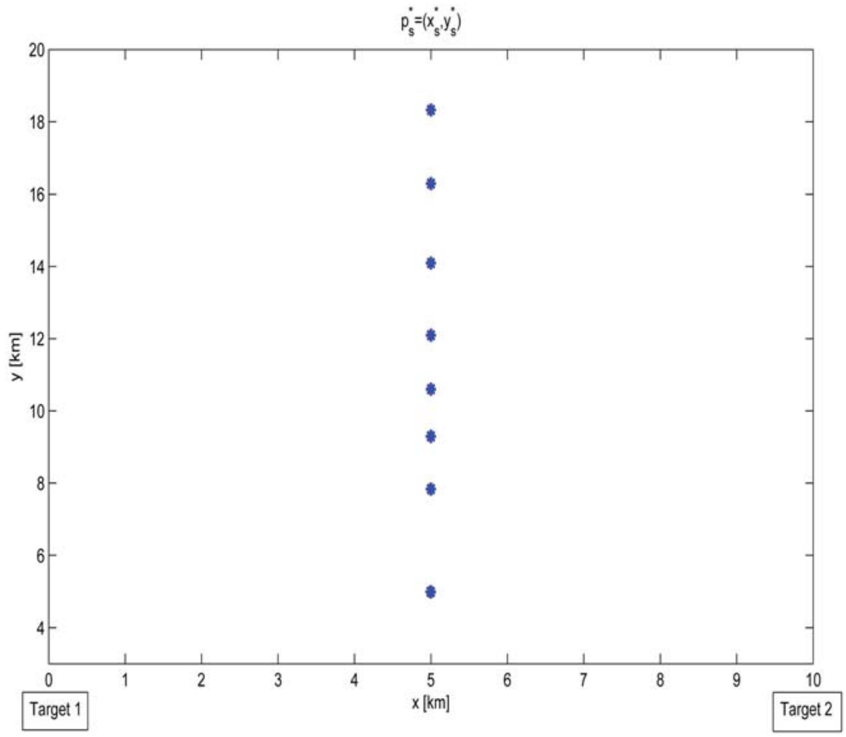


Fig. 5. Pareto optimal locations of the UAV.

In addition, Fig. 5 provides information about the locations of the UAV $p_s^* = (x_s^*, y_s^*)$ that achieve Pareto optimal performance.

4.2. Detection vs. Survivability vs. Tracking

1) *Single target tracking*: This scenario includes tracking a single target t by a UAV s. According to (4) the UAV aims at solving the following two-objective optimization problem

$$\max_{p_s} \begin{bmatrix} \pi_S(\|p_s - p_t\|) \\ \gamma_T(\|p_s - p_t\|) \end{bmatrix}. \tag{14}$$

The parameters of the simulation are as follows. The target is located at $p_t = (10,0)$. The assumed detection

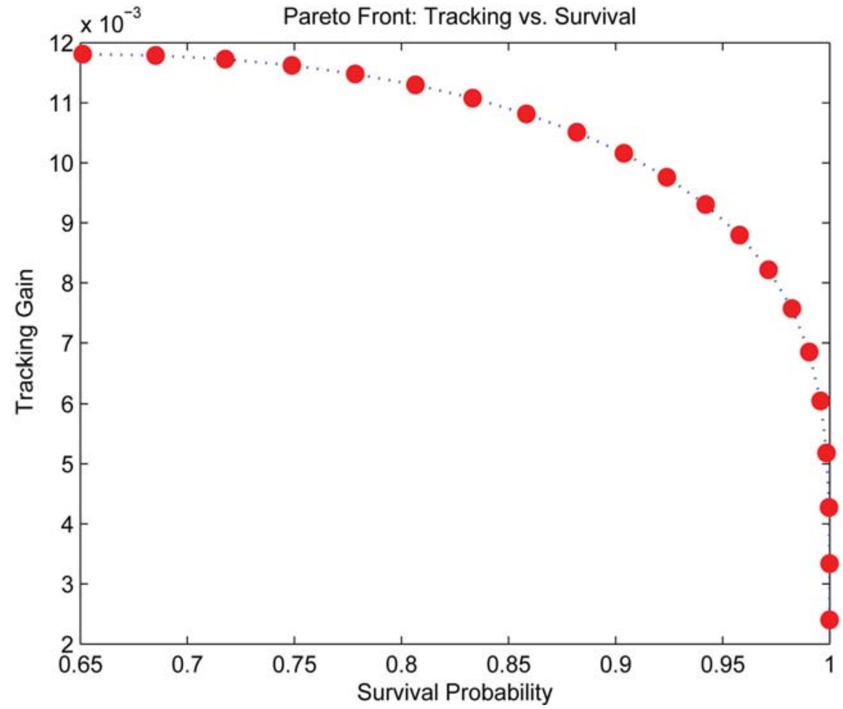


Fig. 6. PF generated by NBI.

and survival functions $\pi_D(d)$ and $\pi_S(d)$ are the same as in Case A.1 (see Fig. 1). It is assumed that the measurement error covariance is $R(d) = (\sigma_{\min}^2 + d^2)\mathrm{I}_2$ with $\sigma_{\min}^2 = 0.1$, and $\bar{I} = \mathrm{I}_4$ where $d = \|p_s - p_t\|$ and I denotes the identity matrix. $H = [\mathrm{I}_2\ \mathrm{O}_2]$ where O denotes the null matrix. Under these assumptions it can be calculated from (3) that

$$\gamma_T(d) = 2\ln\left(1 + \frac{\pi_S(d)\pi_D(d)}{\sigma_{\min}^2 + d^2}\right)$$

where $\pi_S(d)$ and $\pi_D(d)$ are given by (12) and (11), respectively (see Fig. 1).

Fig. 6 shows the obtained Pareto front. Table III gives the fusion weights w_S^* and w_T^* of the WS objective

$$w_S^*\pi_S(\|p_s - p_t\|) + w_T^*\gamma_T(\|p_s - p_t\|)$$

corresponding to the obtained Pareto optimal points $(\pi_S(\|p_s^* - p_t\|), \gamma_T(\|p_s^* - p_t\|))$.

2) *Joint search & tracking*: This scenario includes tracking a single target t and detecting a new target n by a UAV s. The UAV aims at solving the following three-objective optimization problem

$$\max_{p_s} \begin{bmatrix} \pi_D(\|p_s - p_n\|) \\ \pi_S^s(\|p_s - p_n\|, \|p_s - p_t\|) \\ \gamma_T(\|p_s - p_t\|) \end{bmatrix} \tag{15}$$

where under the independent threats assumption

$$\pi_S^s = \pi_S^{s,n}\pi_S^{s,t} = \pi_S(\|p_s - p_n\|)\pi_S(\|p_s - p_t\|)$$

The parameters of the simulation are as follows. The targets' locations are $p_n = (0,0)$ and $p_t = (10,0)$. For the target under track it is assumed that $R = (\sigma_{\min}^2 + \|p_s - p_t\|^2)\mathrm{I}_2$ with $\sigma_{\min}^2 = 0.1$, and $\bar{I} = \mathrm{I}_4$. $H = [\mathrm{I}_2\ \mathrm{O}_2]$. Under these assumptions it can be calculated from (3) that

$$\gamma_T = 2\ln\left(1 + \frac{\pi_S(\|p_s - p_n\|)\pi_S(\|p_s - p_t\|)\pi_D(\|p_s - p_t\|)}{\sigma_{\min}^2 + \|p_s - p_t\|^2}\right)$$

Fig. 7 shows the feasible objective region for detection, survival and tracking, and Fig. 8 shows the obtained Pareto front.

TABLE III
Trade-Off Points & Fusion Weights

π_S	γ_T	w_S	w_T
1.0000	0.0033	0.9013	0.0987
0.9997	0.0043	0.6154	0.3846
0.9985	0.0052	0.3278	0.6722
0.9957	0.0060	0.1742	0.8258
0.9904	0.0068	0.1028	0.8972
0.9824	0.0076	0.0672	0.9328
0.9715	0.0082	0.0475	0.9525
0.9579	0.0088	0.0354	0.9646
0.9420	0.0093	0.0274	0.9726
0.9239	0.0098	0.0218	0.9782
0.9038	0.0102	0.0175	0.9825
0.8819	0.0105	0.0142	0.9858
0.8584	0.0108	0.0115	0.9885
0.8332	0.0111	0.0092	0.9908
0.8065	0.0113	0.0073	0.9927
0.7784	0.0115	0.0056	0.9944
0.7488	0.0116	0.0040	0.9960
0.7177	0.0117	0.0026	0.9974
0.6852	0.0118	0.0013	0.9987

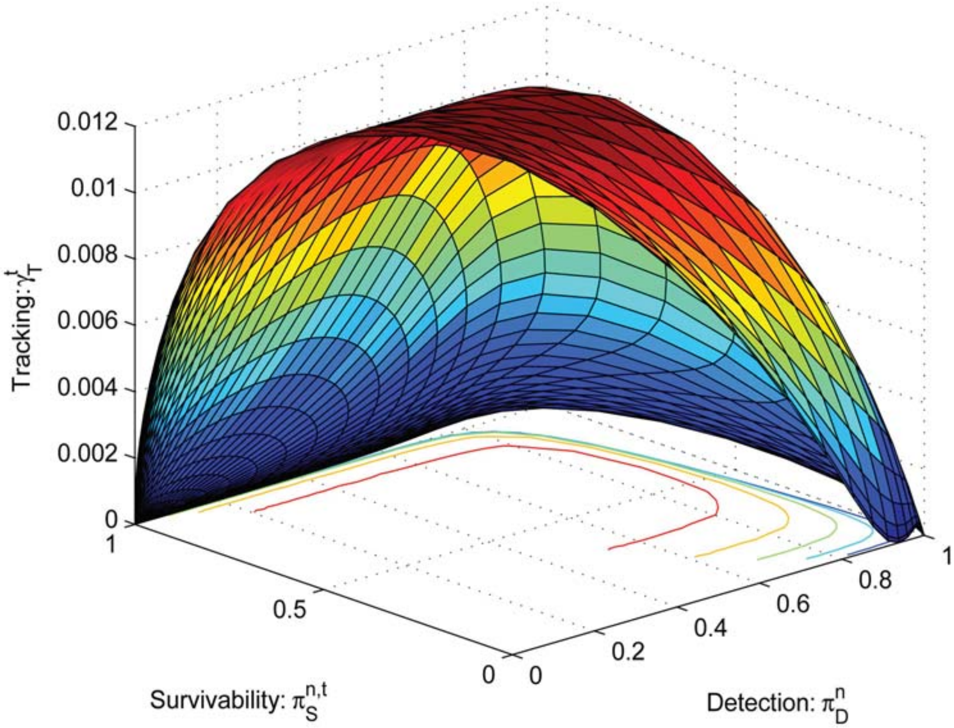


Fig. 7. Feasible objective set.

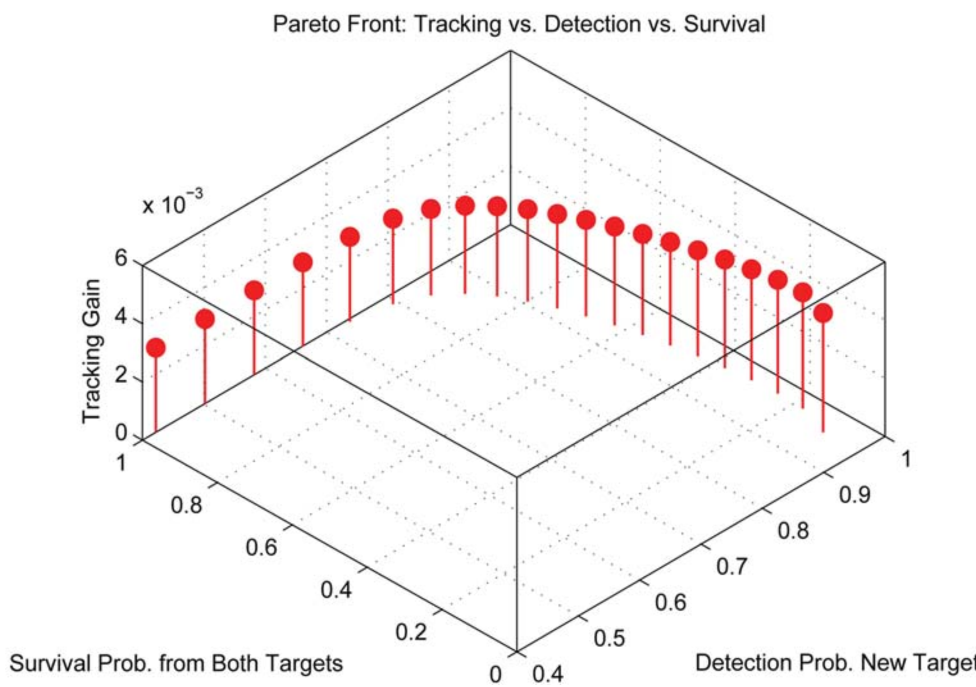


Fig. 8. PF generated by NBI.

Table IV gives the weights w_D^*, w_S^* and w_T^* of the WS objective

$$w_D^* \pi_D(\|p_s - p_n\|) + w_S^{s*} \pi_S^s(\|p_s - p_n\|, \|p_s - p_t\|)$$

$$+ w_T^* \gamma_T(\|p_s - p_t\|)$$

corresponding to the obtained Pareto optimal points $(\pi_D(\|p_s^* - p_n\|),\ \pi_S^s(\|p_s^* - p_n\|, \|p_s^* - p_t\|),\ \gamma_T(\|p_s^* - p_t\|))$.

In addition, Fig. 9 provides information about the locations of the UAV $p_s^* = (x_s^*, y_s^*)$ that achieve the Pareto optimal performance.

5. CONCLUSIONS

A systematic and rigorous multiobjective optimization based approach for designing a fused scalar ob-

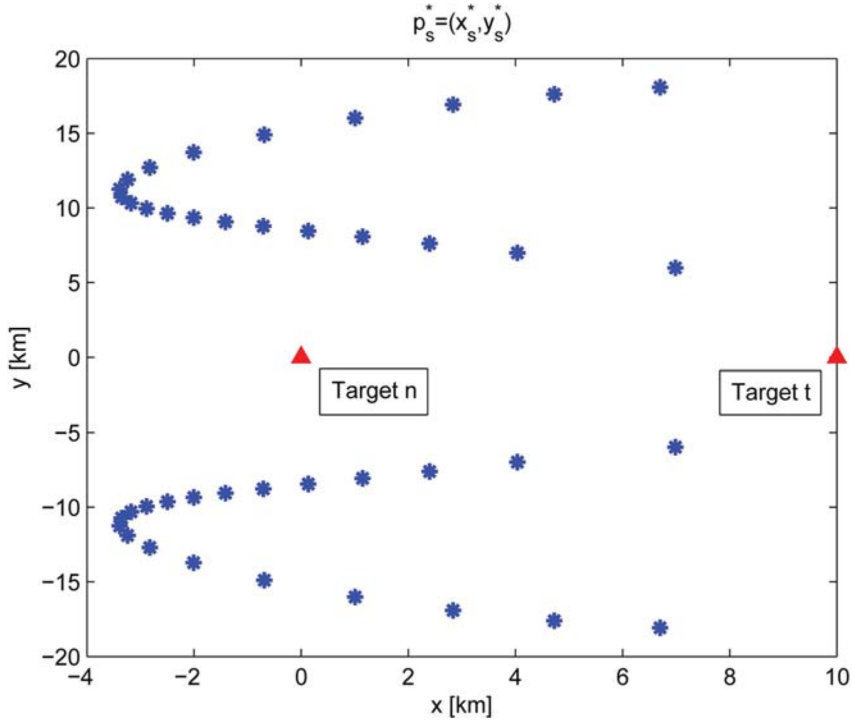


Fig. 9. Pareto optimal locations of the UAV.

TABLE IV
Trade-Off Points & Fusion Weights

π_D	π_S^s	γ_T	w_D	w_S^s	w_T
0.9750	0.2455	0.0039	0.1352	0.0031	0.8617
0.9728	0.3117	0.0038	0.1886	0.0107	0.8007
0.9686	0.3763	0.0037	0.2579	0.0205	0.7216
0.9632	0.4397	0.0036	0.3466	0.0339	0.6195
0.9567	0.5023	0.0036	0.4495	0.0516	0.4989
0.9491	0.5639	0.0035	0.5526	0.0736	0.3738
0.9404	0.6246	0.0034	0.6387	0.0990	0.2623
0.9304	0.6841	0.0033	0.6967	0.1276	0.1757
0.9187	0.7422	0.0032	0.7239	0.1607	0.1154
0.9047	0.7984	0.0032	0.7217	0.2018	0.0765
0.8875	0.8517	0.0031	0.6899	0.2573	0.0528
0.8655	0.9009	0.0030	0.6222	0.3385	0.0393
0.8356	0.9434	0.0030	0.5037	0.4636	0.0327
0.7933	0.9754	0.0029	0.3220	0.6466	0.0314
0.7339	0.9929	0.0029	0.1297	0.8372	0.0331
0.6606	0.9987	0.0029	0.0299	0.9356	0.0345
0.5819	0.9998	0.0029	0.0045	0.9605	0.0350
0.5019	1.0000	0.0029	0.0005	0.9645	0.0350
0.4218	1.0000	0.0029	0.0001	0.9649	0.0350

jective function for search and track missions of unmanned aerial vehicles through weighted combinations of objectives has been proposed. It allows to obtain a representative set of possible trade-off optimal alternatives and determine the weights for the combination of objectives that meets a selected "best" trade-off. The proposed methodology can greatly facilitate the design of mission objective functions through performing insightful trade-off analysis. Its usefulness has been illustrated by results from several case studies of typical search and track mission scenarios.

It should be kept in mind that the method used as well as all numerical methods for general multiobjective optimization can at best provide only *local* Pareto optimality and thus it can be hard sometimes to find initial solutions leading to the trade-off region of practical interest.

REFERENCES

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

[2] I. Das and J. Dennis
A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems.
Structural and Multidisciplinary Optimization, **14**, 1 (1997), 63–69.

[3] I. Das and J. Dennis
Normal-boundary intersection: A new method for generating Pareto optimal points in multicriteria optimization problems.
SIAM Journal on Optimization, **8**, 3 (1998), 631–657.

[4] M. Flint, E. Fernandez-Gaucherand and M. Polycarpou
Cooperative control for UAVs searching risky environments for targets. In *Proceedings of 42nd IEEE Conference on Decision and Control*, vol. 4, Maui, HI, Dec. 2003, 3567–3572,

[5] N. Li and X. R. Li
Target perceivability and its applications.
IEEE Transactions on Signal Processing, **SP-49**, 11 (Nov. 2001), 2588–2604.

[6] K. Miettinen
Nonlinear Multiobjective Optimization.
Boston: Kluwer Academic Publishers, 1999.

[7] K. Passino, M. Polycarpou, D. Jacques, M. Pachter, Y. Liu, Y. Yang, M. Flint and M. Baum
Cooperative control for autonomous air vehicles.
In *Cooperative Control and Optimization*, Kluwer, 2002, 233–269.

[8] M. Polycarpou, Y. Yang and K. Passino
A cooperative search framework for distributed agents.
In *Proceedings of 2001 IEEE International Symposium on Intelligent Control* (ISIC'01), Mexico City, Mexico, Sept. 2001, 1–6.

[9] M. Ridley, E. Nettleton, A. Goktogan., G. Brooker, S. Sukkarieh and H. Durrant-Whyte
Decentralised ground target tracking with heterogeneous sensing nodes on multiple UAVs.
In *Second International Workshop on Information Processing in Sensor Networks*, vol. LNCS 2634, Springer, 2003, 545–565.

[10] A. Sinha, T. Kirubarajan and Y. Bar-Shalom
Autonomous ground target tracking by multiple cooperative UAVs.
In *Proceedings of 2005 SPIE Conference on Signal and Data Processing of Small Targets*, vol. 5913, 2005, 59131V.

[11] A. Sinha, T. Kirubarajan and Y. Bar-Shalom
Autonomous search, tracking and classification by multiple cooperative UAVs.
In *Proceedings of SPIE Signal Processing, Sensor Fusion, and Target Recognition XV*, vol. 6235, May 2006, 623508.

[12] A. Sinha, T. Kirubarajan and Y. Bar-Shalom
Optimal cooperative placement of GMTI UAVs for ground target tracking.
In *Proceedings of SPIE Signal Processing, Sensor Fusion and Target Recognition*, Orlando, FL, Apr. 2004.

[13] R. E. Steuer
Multiple Criteria Optimization: Theory, Computation and Application.
New York: Wiley, 1985.

[14] J. Yan, L. Yan, A. Minai and M. Polycarpou
Balancing search and target response in cooperative unmanned aerial vehicle (UAV) teams.
IEEE Transactions on Systems, Man and Cybernetics, Pt. B, **36**, 3 (June 2006), 571–587.

[15] Y. Yang, A. Minai and M. Polycarpou
Decentralized cooperative search by networked UAVs in an uncertain environment.
In *Proceedings of 2004 American Control Conference*, vol. 6, 5558–5563.

Vesselin P. Jilkov (M'01) received his B.S. and M.S. degree in mathematics from the University of Sofia, Bulgaria in 1982, the Ph.D. degree in the technical sciences in 1988, and the academic rank Senior Research Fellow of the Bulgarian Academy of Sciences in 1997.

He was a research scientist with the R&D Institute of Special Electronics, Sofia, (1982–1988) where he was involved in research and development of radar tracking systems. From 1989 to 1999 he was a research fellow with the Institute of Parallel Processing—Bulgarian Academy of Sciences, Sofia, where he worked as a key researcher in numerous academic and industry projects (including international) in the areas of estimation, target tracking, sensor data fusion, and parallel processing. In 1999 he joined the Department of Electrical Engineering, University of New Orleans, as a visiting scholar and was an assistant professor from 2003 to 2009. At present he is an associate professor. He has been engaged in teaching and conducting research in the areas of hybrid estimation and target tracking. His current research interests include stochastic systems, nonlinear filtering, applied decision & estimation, target tracking, information fusion.

Dr. Jilkov is author/coauthor of over 75 journal articles and conference papers. He is a member of IEEE, ISIF, and SIAM.

X. Rong Li (S'90—M'92—SM'95—F'04) received the B.S. and M.S. degrees from Zhejiang University, Hangzhou, Zhejiang, PRC, in 1982 and 1984, respectively, and the M.S. and Ph.D. degrees from the University of Connecticut, in 1990 and 1992, respectively.

He joined the Department of Electrical Engineering, University of New Orleans in 1994, where he is now University Research Professor, and Director of Information and Systems Technology Research Center. During 1986–1987 he did research on electric power at the University of Calgary, AB, Canada. He was an assistant professor at the University of Hartford, West Hartford, CT, from 1992 to 1994. He has authored or coauthored four books, seven book chapters, and more than 200 journal and conference proceedings papers. His current research interests include signal and data processing, information fusion, target tracking, detection, and classification, statistical inference, performance evaluation, stochastic systems, and electric power.

Dr. Li was elected President of the International Society of Information Fusion as in 2003 and a member of Board of Directors (since 1998); served as general chair for 2002 International Conference on Information Fusion, and steering chair or general vice-chair for 1998, 1999, and 2000 International Conferences on Information Fusion; served *IEEE Transactions on Aerospace and Electronic Systems* as an associate editor (1995–1996) and as editor (1996–2003); served *Communications in Information and Systems* as an editor since 2001; received a CAREER award and an RIA award from the U.S. National Science Foundation. He received 1996 Early Career Award for Excellence in Research from the University of New Orleans and has given numerous seminars and short courses in North America, Europe, Asia, and Australia. He won several outstanding paper awards, is listed in *Marquis' Who's Who in America* and *Who's Who in Science and Engineering*, and consulted for several companies.

T2T and M2T Association with Combined Hypotheses

JAVIER ARETA
YAAKOV BAR-SHALOM
KRISHNA R. PATTIPATI

This paper presents a procedure to combine the top association hypotheses generated in a track-to-track (T2TA) association problem. The standard procedure for such problems consists of keeping only the most likely hypothesis, but the extra information carried by other hypotheses remains unused. The proposed combination method allows for the extraction of this information in an efficient way, improving over a similar method [5], providing system tracks that account for the correlation ambiguity. This method will prove useful when there is track contention (correlation ambiguity), and the information carried by the best hypothesis alone renders optimistic estimates. As a result of using this method, both better estimates (fused system tracks) are obtained and an estimate of the difficulty of the association problem is obtained based on the aggregation of neighboring tracks. In this work we consider two applications, one consisting of a T2T fusion (T2TF) and a dynamic tracking problem where measurement-to-track association (M2TA) hypotheses from a multiple hypothesis tracker (MHT) are combined. The comparison of results from the proposed procedure vs. the standard approach indicate that the latter can be improved upon in scenarios with significant association ambiguities.

Manuscript received January 25, 2009; released for publication May 21, 2009.

Refereeing of this contribution was handled by Stefano Coraluppi.

Authors' address: University of Connecticut, ECE Department, Storrs, CT 06269-2157, E-mail: (areta@ee.uconn.edu; ybs@ee.uconn.edu; krishna@ee.uconn.edu).

1557-6418/09/$17.00 © 2009 JAIF

1. INTRODUCTION

The types of problems in data association for tracking are (i) measurement-to-measurement association (M2MA), i.e., track initiation, (ii) measurement-to-track association (M2TA), i.e., track continuation, (iii) track-to-track association (T2TA), for track fusion. Among the M2TA algorithms, it is well known that the Multiple Hypothesis Tracker (MHT) performs, due its time window, much better than other methods, such as nearest neighbor or PDA (which have a time window of depth 1) when there is heavy clutter or track ambiguity, i.e., when tracks are very close or cross each other. The idea behind MHT is to maintain several track-to-measurement association hypotheses over its time window, some of which may have low likelihood but might later become the most likely after some frames of measurements have been added. In general, however, only the best hypothesis is retained when obtaining the results of the tracker at a particular time, thus neglecting the information contained in the subsequent hypotheses. In the T2TA problem, the use of the approach [3] yields only the most likely association, in a manner similar to the MHT.

This paper presents methods to combine the top hypotheses generated in M2TA and T2TA problems, extending the results from [5] which proposed the Coordinated Presentation (CP). The T2TA problem consists of estimating the parameters of interest for an unknown number of targets, using the track lists obtained by S observers, which are received by a fusion center (FC). The fusion center generates several association hypotheses, each of them formed by associating tracks (the list elements) into S-tuples, using an m-best multidimensional assignment (MDA) algorithm based on Lagrangean relaxation [8]. The goal is to combine those hypotheses to obtain a better estimate than the one calculated using the top hypothesis alone. The best hypothesis estimate has the disadvantage of being optimistic, especially when there is track ambiguity, i.e., tracks are close to each other relative to their covariances (small normalized distance). For example, if the second best hypothesis has a likelihood close to the best, it should be accounted for: the covariances calculated assuming the best hypothesis is guaranteed to be true are optimistic because the second best might be the true one. The use of the top m hypotheses to asses the quality of the association was proposed in [6]. There the best assignment is used to update only if all of its association S-tuples are substantially present in the subsequent hypotheses. If some of them do not appear in subsequent hypothesis with high probability, an extended window is used to hopefully clear up the problem. Our approach is to combine the hypotheses to avoid incurring any delay.

The M2TA problem considered requires the assignment of noisy measurements from a single radar arising from N targets. At time k window containing S-1 frames of data (lists of measurements) and the list of accepted tracks from time $k - S + 1$, i.e, a total of S lists is used to generate the m-best association hypotheses to be combined. After the new track estimates are obtained, the accepted tracks are extended to the next frame $(k - S + 2)$, and a new frame (from time $k + 1$) is incorporated, and the process repeated with a total of S lists, shifted one step forward.[1]

The paper is organized as follows: Section 2 discusses the procedure to combine the hypotheses in a T2TA problem, and compares it to the method in [5]. Section 3 presents a track-to-track association scenario and shows the results of applying the algorithms discussed previously. M2TA is considered in Section 4. Finally, in Section 5 conclusions are drawn based on the results obtained.

2. COMBINATION OF THE m-BEST HYPOTHESES IN T2TA

Each of the m hypotheses obtained by the MDA algorithm is formed by N_i, $i = 1,\ldots,m$ S-tuple associations, which we will simply call 'associations.' Our goal is to combine these top hypotheses in a combined hypothesis (discussed in detail in Sec. 2.2, from which the combined (system) track estimates are better than the best hypothesis alone.

The hypothesis combination consists of two subproblems:

- Finding similar associations (deemed to represent the same target) from different hypotheses to form *combinable association sets* (called $\mathcal{C}$-sets for brevity), which is a data association/combination problem.
- Combination of these sets into a unique estimate, to provide both an estimate of the number of targets present, as well as estimates of the parameters of interest and their covariances.

To be able to weight the contribution of each hypothesis to the final estimate, the probability of a hypothesis needs to be computed. This probability is obtained from the costs of the hypotheses. The cost of hypothesis i is the sum of the costs of the N_i associations that comprise it. These association costs are based on the negative log likelihood ratio (NLLR), that is, the negative of the logarithm of the likelihood function of the set of tracks from an association having common origin divided by the likelihood function of not having a common origin, given by

$$\ell_i \triangleq -\ln \mathcal{L}(H_i) = -\sum_{j=1}^{N_i} \ln(\Lambda_{ij}/\Lambda_{0j}), \qquad i = 1,\ldots,m \tag{1}$$

where $\mathcal{L}(H_i)$ is the LR of H_i, Λ_{ij} is the likelihood function of the association tuple j from hypothesis i having common origin, and Λ_{0j} is the likelihood function of this tuple not having common origin. The probability of hypothesis i is obtained by normalization of these costs as

$$P\{H_i\} = \frac{e^{-\ell_i}}{\sum_{j=1}^{m} e^{-\ell_j}}, \qquad i = 1,\ldots,m. \tag{2}$$

For this to be meaningful, the probability of the mth hypothesis should be much smaller than the highest one, so the hypotheses left out can be deemed to have negligible probability.

Thus, for purposes of association combination, each association j in hypothesis i is assigned two numbers:

1. A value corresponding to the association weight equal to $P\{H_i\}$, to be used to quantify the number of tracks in each $\mathcal{C}$-set.
2. A value corresponding to the probability of an association, called combination weight. This value is defined in a similar way to the probability of a hypothesis (2) within a $\mathcal{C}$-set, but using the NLLR of the association A_{ij} instead of the likelihood of the complete hypothesis, as detailed below in (4). This probability is needed at the fusion center for weighting the contribution of each association in a $\mathcal{C}$-set to the overall estimate from such a set.

The association weights are used to define the total probability of a $\mathcal{C}$-set. For each $\mathcal{C}$-set an indicator function is defined, which is one when association A_{ij} is included in $\mathcal{C}_k$, namely,

$$\chi_{ij}(\mathcal{C}_k) = \begin{cases} 1 & A_{ij} \in \mathcal{C}_k \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

Using this indicator function, the total probability of a $\mathcal{C}$-set is defined in terms of the association weights as

$$P\{\mathcal{C}_k\} = \sum_{i,j} \chi_{ij}(\mathcal{C}_k) P\{H_i\}. \tag{4}$$

2.1. $\mathcal{C}$-set Generation

The criterion for combining associations in a $\mathcal{C}$-set is based on the number of common reports (tracks in the T2TA case, measurements in the M2TA case) shared by associations in the set of all associations

$$\mathcal{A} = \{A_{ij},\ i = 1,\ldots,m,\ j = 1,\ldots,N_i\}. \tag{5}$$

This follows the reasoning from the Coordinated Presentation (CP) approach [5]. In that work a *minimal similarity criterion* is used, i.e., if two associations share one or more elements (tracks or measurements), they are deemed as coming from the same origin and thus are included in the same $\mathcal{C}$-set. This approach will be shown to be prone to incorrect combinations.

[1] In T2TA, S is the number of observers, while in M2TA S is the time depth of the window. In both cases one has S lists and the dimension of the assignment is S.

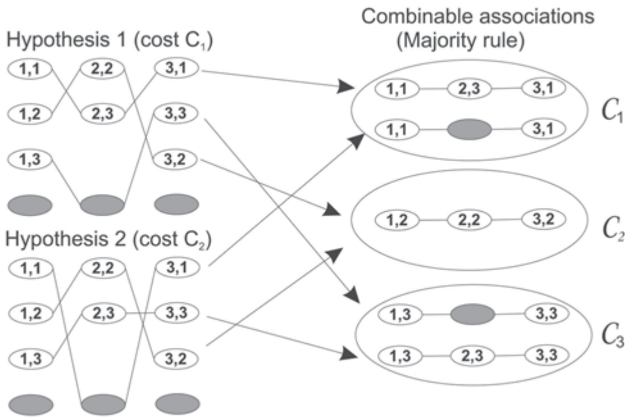


Fig. 1. Associations from 2 hypotheses in a 3 target and 3 sensor case (solid elements correspond to dummies—missing elements in an association). List elements are indexed by number pairs corresponding to (sensor, element). $\mathcal{C}$-set formation criterion is majority rule.

In the present work, the rules used for defining $\mathcal{C}$-sets are:

- Each of the associations in the best hypothesis initializes a $\mathcal{C}$-set.
- Associations are added to these $\mathcal{C}$-sets using a K-*similarity criterion*, that is, if they share at least K common elements with an association already in the $\mathcal{C}$-set. In general K will be taken as $\lfloor S/2 \rfloor + 1$, where S is the number of lists, thus the criterion reduces to a *majority rule criterion*.
- If an association passes the K-similarity criterion for more than one $\mathcal{C}$-set, these are merged into a single one including all the associations in the overlapping $\mathcal{C}$-sets.

Other rules for inclusion to a $\mathcal{C}$-set can be used, as normalized distance between tracks, or normalized distance with respect to a centroid, but are not considered in the sequel.

Fig. 1 shows a simple case with 2 hypotheses, where 3 targets are present and 3 sensors provide tracks. If the $\mathcal{C}$-sets are formed based on associations that satisfy the majority rule, then there will be 3 such sets. Only one of them, $\mathcal{C}_2$ is formed by the same association in each hypothesis, the other are composed of hypotheses that share 2 elements with common origin.

2.2. Combination in a $\mathcal{C}$-set

2.2.1. Estimation of the number of targets in a $\mathcal{C}$-set

Once the $\mathcal{C}$-sets are generated, the combination of the associations in each of them needs to be done. Also, the estimation of the number of true target(s) from which the tracks in each $\mathcal{C}$-set originated is performed, using the total probability of the hypotheses in the $\mathcal{C}$-set. Possible situations are

1. Exactly the same association is present in all hypotheses. In this case a set will contain only this association, and the total probability of such set will be $P\{\mathcal{C}_k\} = \sum_{i=1}^{m} P\{H_i\} = 1$, indicating that all the associations correspond to a unique target. Such a case indicates the high quality of the association, and hence it can be trusted and presented as is.
2. At least one different association satisfying the common set inclusion condition is present in each hypothesis. In this case, the $\mathcal{C}$-set will contain different associations, and the total probability $P\{\mathcal{C}_k\}$ of such a set can be greater than 1, indicating that the combined estimate represents closely spaced tracks.[2]
3. Not all the hypotheses contribute to a $\mathcal{C}$-set. In this case, the total probability of the $\mathcal{C}$-set will be smaller than 1. This can happen when there are associations containing false tracks, or few tracks, thus having not enough in common with established sets.

In the second case, some associations will K-overlap some other associations, and two or more associations from the same hypothesis may be included in the set. If the total probability of the set is higher than 1, this indicates that there is more than one true track in such a set, which now represents a *cluster of targets*. This cluster is considered to contain as many targets as the (rounded up) total probability $P\{\mathcal{C}_k\}$. A unique estimate that plays the role of a centroid can be obtained from such a cluster by combining the estimates arising from each association in it. If the total probability is close to 1, the combination of the associations in the cluster should provide a better estimate of the target it represents, compared to the one provided by the best hypothesis alone. This is discussed in more detail later.

2.2.2. Combination of associations in a $\mathcal{C}$-set

The combination of the associations in a $\mathcal{C}$-set can be done using the CP method from [5], which we call the Coordinated Presentation (CP) Mixture Approach. This combines the estimates of these association using the association weighting to define the combination probability. Our approach, called Direct Mixture (DM), will use a different—likelihood ratio based—weighting, defined before as combination weighting.

CP Mixture (CP) Approach

For each association in a $\mathcal{C}$-set the following events are defined:

$$A_i = \{\text{set } X_i \text{ of } n_i \text{ tracks are associated}\} \tag{6}$$

where $X_i = \{\hat{x}^{i_j}, P^{i_j}\}_{j=1}^{n_i}$ is the set of estimates and covariances contained in association i. Suppose that the set of all the N associations in a $\mathcal{C}$-set A is

$$A = \{A_i,\ i = 1,\ldots,N\}. \tag{7}$$

[2]Suppose two hypotheses such that $P\{H_i\} = .5$, with two associations each. H_1 contains $\{M_1^1–M_2^2\}$ and $\{M_2^1–M_1^2\}$, while H_2 contains $\{M_1^1–M_1^2\}$ and $\{M_2^1–M_2^2\}$, where M_i^j corresponds to element i of list j. If the set inclusion condition is to have at least 1 common element, then the $\mathcal{C}$-set will be $\{\{M_1^1–M_2^2\},\{M_2^1–M_1^2\},\{M_1^1–M_1^2\},\{M_2^1–M_2^2\}\}$, which has $P\{\mathcal{C}\} = 2$.

The mixing probability p_i for association event A_i is taken as the probability of the hypothesis k_i it comes from, $P\{H_{k_i}\}$, as defined before.

For each association event A_i, the combined "system" state estimate and covariance are $\hat{x}^i$ and P^i. The combination of these estimates is done using the mixture pdf

$$p(x \mid A) = \frac{1}{\sum_{j=1}^{N} p_j} \sum_{i=1}^{N} p(x \mid A_i) p_i. \tag{8}$$

The mean and covariance of this mixture are

$$\hat{x} = \frac{1}{\sum_{j=1}^{N} p_j} \sum_{i=1}^{N} \hat{x}^i p_i \tag{9}$$

$$P_{\hat{x}} = \frac{1}{\sum_{j=1}^{N} p_j} \sum_{i=1}^{N} (P^i + \hat{x}^i (\hat{x}^i)') p_i - \hat{x}\hat{x}'. \tag{10}$$

Direct Mixture (DM) Approach

Using the previous definition of event A_i, the likelihood ratio (LR) of event A_i can be written as

$$\mathcal{L}(A_i) = \frac{\Lambda(A_i)}{\mu^{n_i - 1}} \tag{11}$$

where $\Lambda(A_i)$ is the likelihood function (LF) of association event A_i—that X_i have common origin—and μ^{n_i-1} is the LF of not having common origin [3]. The term μ is the density of extraneous measurements, defined as n_{ex}/V, where n_{ex} is the number of extraneous tracks, and V is the volume of the state space corresponding to the surveillance region.

The mixing probability of association i within its $\mathcal{C}$-set can be taken as

$$\tilde{p}_i = \frac{1}{c} \mathcal{L}(A_i) \tag{12}$$

where

$$c = \sum_{i=1}^{n} \mathcal{L}(A_i). \tag{13}$$

Using (dimensionless) likelihood ratios makes it possible to have associations with different n_i in a $\mathcal{C}$-set.

For each association we can obtain the combined (system) state estimate and covariance

$$\hat{x}^i = \varphi(X_i \mid A_i) \tag{14}$$

$$P^i = \Phi(X_i \mid A_i) \tag{15}$$

according to [3].

The combination of these estimates is also done using a mixture, with different weights $\tilde{p}_i$ based on their individual likelihood ratios (12), rather than based on the hypotheses probabilities. The mixture pdf is

$$p(x \mid A) = \sum_{i=1}^{N} p(x \mid A_i) \tilde{p}_i \tag{16}$$

with mean and covariance

$$\hat{x} = \sum_{i=1}^{N} \hat{x}^i \tilde{p}_i \tag{17}$$

$$P_{\hat{x}} = \sum_{i=1}^{N} (P^i + \hat{x}^i (\hat{x}^i)') \tilde{p}_i - \hat{x}\hat{x}' \tag{18}$$

where the estimates coming from the combination of X_i in (14),(15) are given by[3]

$$P^i = \left(\sum_{j=1}^{N} (P^{i_j})^{-1} \right)^{-1} \tag{19}$$

$$\hat{x}^i = P^i \left(\sum_{j=1}^{N} (P^{i_j})^{-1} \hat{x}^{i_j} \right). \tag{20}$$

3. SIMULATION RESULTS

The results presented in this section correspond to the track-to-track association problem from [1], where missile launch event parameters are estimated at a fusion center collecting track reports from several observers. The scenario 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 FC has to associate the common origin (same event) tracks from each observer, then associate and fuse the most recent common origin tracks across the observers. For more details, see [1].

3.1. Problem Description

The parameters to be estimated are positions in x, y, heading and launch time. The surveillance area is $[0, 10000]$ distance units both in x and y, $[0, 30]$ degrees for the heading, and $[0, 10]$ units of time for the launch time. We assume that the sensors used have enough spatial resolution to detect the events as unique, and also that the time difference between launches is enough to recognize them as different events. Track/event reports are produced by 4 observers, and are transmitted to a fusion center for processing. These reports are based on observations corrupted by Gaussian noise with a standard deviation of 800 in both x and y coordinates, 10 degrees for the heading and 3 units of time for the launch time. Fusion is performed every 20 units of time, over a time span of 200 time units. The initial frames of data (reports) have larger variance (because they are based on fewer measurements) than the latter ones, which are about 5 times more accurate due to the processing of

[3]Assuming as in [1] that the track errors are uncorrelated.

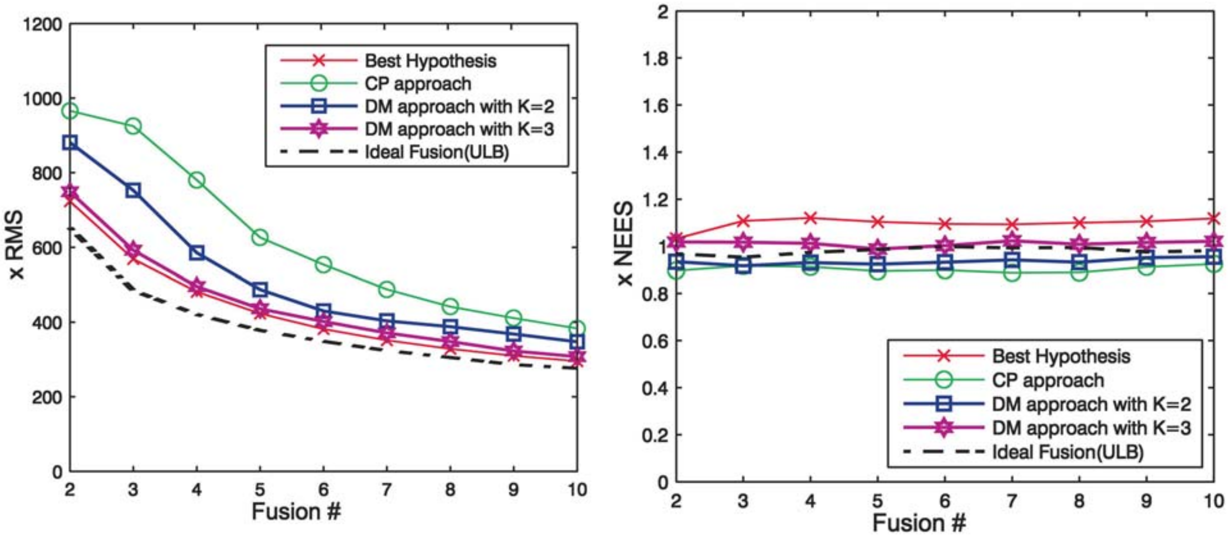


Fig. 2. RMS and NEES in the x coordinate. The normalization is done w.r.t the obtained covariance from the fuser for each estimate.

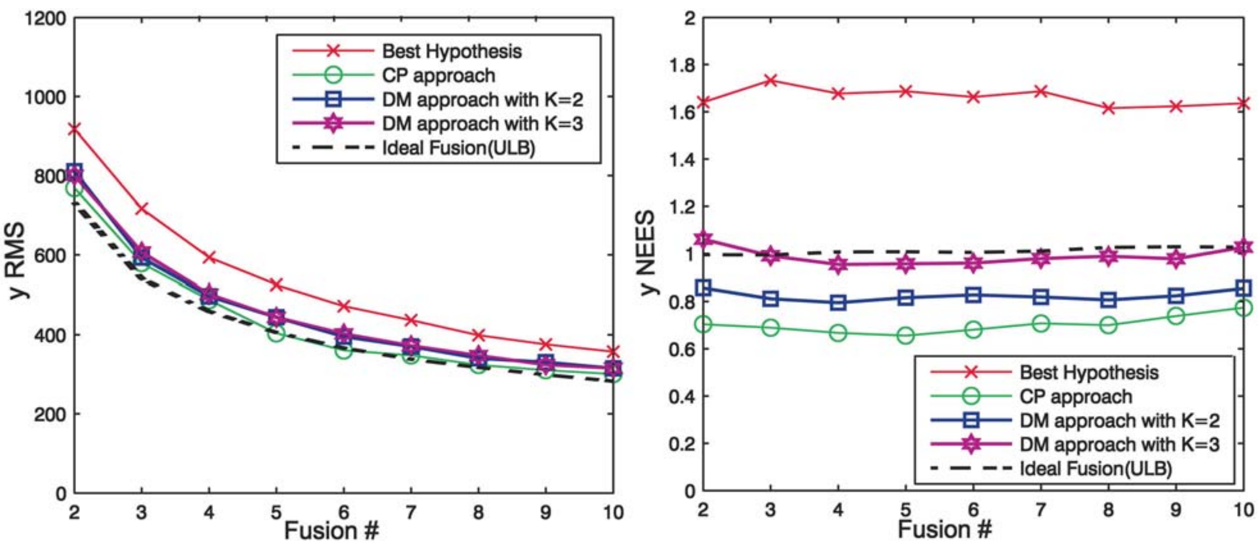


Fig. 3. RMS and NEES in y coordinate. The normalization is done w.r.t the obtained covariance from each estimate.

more measurements. Thus, we expect more tracks contending at the beginning and this contention should be resolved for most cases by the final time.

Results shown are based on 500 Monte Carlo runs. In each run, the RMS error is calculated for the CP and DM association algorithms using the combined m-best hypotheses, for the single best hypothesis, as well as for the true association (known only in simulations) to be used as a reference. Also the normalized estimation error squared (NEES) [3] is obtained for each. The total number of $\mathcal{C}$-sets and the number of $\mathcal{C}$-sets with total probability 1 will be used to quantify the degree of aggregation of the track estimates.

3.2. Scenario

We consider a scenario with 8 launches uniformly distributed in a line parallel to the x axis. The distance between neighboring launches is 800 meters. For this case, missasociation and clustering is expected to happen often for the initial time, while towards the end time this effect should decrease. Results for the CP approach as well as our DM approach with $K = 2,3$ are shown in Figs. 2 and 3 for the x and y coordinates, respectively. These figures contain five curves, one corresponding to the best hypothesis, a second one that corresponds to doing the ideal association, i.e., using the true association indexes,[4] and other three corresponding to the m-best hypotheses combination methods.

From the NEES values it can be seen that for this problem the best hypothesis estimate is optimistic for both coordinate estimates.

The geometry of the problem results in different behavior for the estimates in x and y coordinates. For the y coordinate estimation, the aggregation of tracks does improve the estimate, as these tracks have varia-

[4]This curve corresponds to the unattainable lower bound (ULB).

TABLE I
Average Number of $\mathcal{C}$-sets and Average Total Number of Targets Reported by Each Method

	# of single target $\mathcal{C}$-sets/# of cluster $\mathcal{C}$-sets/# of targets								
Fusion	2	3	4	5	6	7	8	9	10
Ideal	NA/NA 6.64	NA/NA 7.68	NA/NA 7.76	NA/NA 7.84	NA/NA 7.84	NA/NA 7.92	NA/NA 7.92	NA/NA 8.00	NA/NA 8.00
CP	0.82/1.02 6.29	0.91/1.04 7.84	1.22/1.08 7.94	1.30/.98 7.96	1.38/1.10 7.97	1.32/1.05 8.0	1.44/1.12 8.0	1.75/1.12 8.0	1.87/1.13 8.0
DM with $K = 2$	1.12/1.23 6.24	1.28/1.21 7.76	1.04/1.12 7.92	1.34/1.07 7.92	1.77/1.23 7.92	1.98/1.17 8.0	2.15/1.51 8.0	2.84/1.27 8.0	3.11/1.37 8.0
DM with $K = 3$	2.42/1.18 6.56	2.54/1.73 7.84	3.21/1.45 7.84	3.56/1.41 7.92	3.63/1.85 8.0	3.88/1.15 8.0	4.01/1.42 8.0	4.23/1.27 8.0	4.40/1.36 8.0

tion around the same value. On the other hand, for the estimates in the x coordinate, the aggregation decreases the quality of the estimate, as the different tracks combined possess different x coordinate values. In Fig. 2 the error in x is shown for all the methods considered. It can be seen that the CP approach does provide very inaccurate x estimates as a result of its loose set inclusion condition, which forces the combination of insufficiently similar tracks in the same $\mathcal{C}$-set. The proposed DM combination scheme using $K = 2$ does improve over the CP case, but only marginally. The more stringent condition $K = 3$ provides enough discrimination so as to reduce the RMS error to the level of the top hypothesis alone (but with consistency as good as the ideal). Overall, the methods provide consistent (or nearly so) covariance calculations, as can be seen from the normalized error plots. Fig. 3 shows the error in estimation of the y coordinate values. As opposed to the x estimates, all the methods using the top m hypotheses provide good estimates, as explained before, although the CP method does report a slightly pessimistic variance estimate. On the other hand, the top hypothesis alone does not only provide a more inaccurate estimate, it also lacks consistency, reporting very optimistic results, up to 80% off from its correct value.

It can be concluded that the proposed DM combination method with $K = 3$ is able to combine similar hypotheses effectively and keep enough unique track sets. As a result, the estimates in the x coordinates are as good as the ones provided by the top hypothesis alone, and much better than the estimates obtained by the other two combination schemes. For the y coordinate estimates both the estimation accuracy as well as consistency are improved over the top hypothesis alone, while the results for the other two schemes are comparable.

Table I shows the number of $\mathcal{C}$-sets with unity total probability,[5] i.e, single target, and the number of cluster $\mathcal{C}$-sets (with multiple targets), as well as the total number of targets declared by each method.

[5]These are the $\mathcal{C}$-sets with total probability between 0.5 and 1.5.

4. COMBINATION OF THE m-BEST HYPOTHESES IN MHT

The M2TA combination problem is very similar to the T2TA combination problem, differing mainly in the way the association costs are calculated. In this problem, the first list in the sliding window MHT implementation consists, at the current time k, of the track estimates at time $k - S - 1$ and the following S-1 lists of measurements from frames $k - S + 2, \ldots, k$. Thus, each of the S-tuple associations is formed by a track and S-1 measurements. The incremental cost of such an association is obtained based on the innovation and innovation pdf of the last measurement. The likelihood ratio for continuation of track t with measurement z_j is [2]

$$L_{tj} = \frac{f_t(z_j(k))}{\lambda_\theta} P_{D_t}(k) \tag{21}$$

where $f_t(z_j(k))$ is the pdf of the predicted measurement for track t and λ_θ represents the density of extraneous measurements.

The likelihood ratio in case a measurement is not associated to track t is

$$L_{t0} = 1 - P_{D_t}(k). \tag{22}$$

Thus, considering independent measurement errors, the cost of an association is obtained by summing up the negative of the logarithm of the likelihood ratios involved, calculated sequentially by updating the track from frame 1.

After the m-best hypotheses have been calculated, the $\mathcal{C}$-set generation is done following the same procedure outlined in Subsection 2.1. Associations from different hypotheses sharing K elements are to be combined, and an estimate of the number of tracks in the obtained centroid can be calculated based on the hypothesis probability.

4.1. Track Combination

The combination of track estimates, which will be used in the following cycle of the algorithm, requires careful analysis. The concept is that out of the S-tuple associations to combine, only the estimates corresponding to the first measurement to track association should be taken into account. These combined tracks, originating from elements in the first and second frame, form the new (combined) track estimate, with combination weights based on the cost of the complete association. A better understanding of the combination is obtained by looking first at the way information is processed in the conventional MHT, and then looking at the proposed approach.

4.1.1. Conventional Implementation

From the initial time, the tree of associations is expanded until time S (except for applying steps to reduce the number of branches of the tree). At time S one has the most probable hypothesis, designated by the index $l^*(S)$, as

$$\Theta^{S,l^*(S)} \triangleq \Theta^{[1,S],l^*(S)} = \{\theta(1)^{l^*(S)}, \Theta^{[2,S],l^*(S)}\} \qquad (23)$$

written decomposed into its initial part $\theta(1)^{l^*(S)}$ from time 1, and its part $\Theta^{[2,S],l^*(S)}$ from the interval $[2,S]$.

At $k = S+1$ the hypotheses to be considered are

$$\Theta^{S+1,l} = \{\theta(1)^{l^*(S)}, \Theta^{[2,S+1],l}\} \qquad (24)$$

i.e., all the hypotheses have a frozen common root behind the window $[2,S+1]$, of length S.

In general, at time $k > S$, the hypotheses to be considered are

$$\Theta^{k,l} = \{\Theta^{[1,k-S],l^*(k-1)}, \Theta^{[k-S+1,k],l}\} \qquad (25)$$

i.e., behind the window $[k-S+1,k]$, all the associations are frozen.

4.1.2. Implementation with Combined Hypotheses

The new alternative is to use a *combination* of the hypotheses behind the window, rather than only the most likely one.

This is accomplished as follows. Let the combined hypothesis at time 1 based on the data at time S be

$$\bar{\theta}(1) \triangleq \bar{\theta}[\theta(1)^{i,S},\ i = 1,\ldots,n_1] \qquad (26)$$

where $\theta(1)^{i,S}$ is hypothesis i based on the data at time S, and n_1 is the number of hypotheses at time 1 used in the combination.

This leads to the set of track estimates at time 1

$$X(1 \mid 1) \triangleq \{\hat{x}^t[1 \mid 1, \bar{\theta}(1)]\}_{t=1}^{n_t(1)} = \kappa[\hat{x}[1 \mid 1, \theta(1)^{i,S}],\ i = 1,\ldots,n_1] \qquad (27)$$

where t is the target index and κ is the combination function. This will become the sufficient statistic (initial condition) to be used in forming the new hypotheses in the next window $[2,S+1]$.

At time $S+1$, the hypotheses can therefore be written as

$$\Theta^{S+1,l} = \{X(1 \mid 1), \Theta^{[2,S+1],l}\} \qquad (28)$$

which replaces (24). Similarly, for general $k > S$, (25) is replaced by

$$\Theta^{k,l} = \{X(k-S \mid k-S), \Theta^{[k-S+1,k],l}\}. \qquad (29)$$

The resulting algorithm is designated as Top m Hypotheses Tracker (TmHT).

4.2. Simulations

The results presented in this section correspond to a simple M2TA problem where $N = 2$ targets move in formation, following parallel straight line trajectories. The purpose of the simulations is to compare the performance when the targets are close to each other, such that the measurements from one of them are likely to be confused as originated from the target of interest.

A nearly constant velocity (NCV) motion model based on a discretized continuous time white noise acceleration (CWNA) model is used to characterize the dynamics of the target [4], namely,

$$x(k+1) = Fx(k) + v(k) \qquad (30)$$

where

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (31)$$

The discrete time noise $v(k)$ has covariance matrix

$$Q = \begin{bmatrix} T^3/3 & 0 & T^2/2 & 0 \\ 0 & T^3/3 & 0 & T^2/2 \\ T^2/2 & 0 & T & 0 \\ 0 & T^2/2 & 0 & T \end{bmatrix} \tilde{q} \qquad (32)$$

where $\tilde{q}$ is the power spectral density (psd) of the continuous time zero-mean white process noise that models possible target maneuvers.

The measurement vector z consists of x and y elements,

$$z(k) = Hx(k) + w(k) \qquad (33)$$

where

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad (34)$$

with measurement noise covariance matrix R. The targets are considered to travel in parallel trajectories with velocity $v_x = 0$ in the x direction and $v_y = 5$ in the y direction. The separation between the targets is parameterized by c, the distance in the x direction, while the distance in y is taken to be 0. The sampling time is $T = 1$, and the measurement noise is zero mean i.i.d. Gaussian with standard deviations $\sigma_x = 4$, $\sigma_y = 2$. The value of the density of extraneous measurements used

TABLE II
RMS Error for x and y Coordinates for the 3 Association Methods, as a Function of Different Separations c (in the x direction), using a Window of $S = 3$ Frames

	x			y		
c	TmHT-CP	TmHT-DM	MHT	TmHT-CP	TmHT-DM	MHT
14	6.36	3.45	3.70	1.19	1.50	1.31
12	5.77	3.34	4.16	1.14	1.48	1.48
10	5.33	2.78	4.53	1.06	1.45	1.48
8	4.43	2.81	5.32	1.02	1.28	1.51

TABLE III
RMS Error for x and y Coordinates for the 3 Association Methods, as a Function of Different Separations c (in the x direction), using a Window of $S = 4$ Frames

	x			y		
c	TmHT-CP	TmHT-DM	MHT	TmHT-CP	TmHT-DM	MHT
14	6.07	2.48	2.50	1.18	1.47	1.46
12	5.59	3.03	3.41	1.14	1.47	1.45
10	5.26	3.19	4.91	1.05	1.46	1.46
8	4.49	3.16	5.29	1.02	1.30	1.51

is proportional to the inverse of the square root of the volume of the measurement noise covariance matrix, $\lambda_\theta = c_1\left(\pi\sqrt{\det(R)}\right)^{-1}$. The value of c_1 used is 0.05. For simplicity it is assumed that the target detection probability is unity. The approach is applicable to detection probability less than unity using the likelihood ratio as in (21).

The three aforementioned algorithms are used to track the targets for each of the possible separations. The first two pertain to the family of Top m MHT (TmHT) algorithms, namely the CP approach (TmHT-CP) and the DM approach using the majority rule (TmHT-DM), while the third is the conventional MHT, which retains only the most likely hypothesis at the rear end of the sliding window. RMS errors are obtained from 100 Monte Carlo runs for each of the methods, and for two different number of frames in the sliding window (the time depth), $S = 3,4$.

Tables II and III present the RMS errors in x and y coordinates for the cases of using windows of $S = 3$ and $S = 4$ lists, respectively. The more important results correspond to the x coordinate, as it is the resolvable one, and the one for which the separation is varied. For the case $S = 3$ it can be seen that CP is a simplistic approach, and that coalescence is too prone to happen, thus loosing track accuracy. The MHT gives good results when track separation is larger, but its performance degrades as the targets get close to each other, due to track switching. The DM approach performs better than both of them, at the expense (when compared to MHT) of losing the track ID whenever tracks coalesce. This indeed may be an advantage, as it is preferable to know that a region of potential confusion arised, making the tracker lose the target IDs, rather than keeping the potentially wrong IDs.

The results for the y coordinate are very similar for all separations in this case (as well as for the case $S = 4$), and provide little information as the targets have the same y coordinate position at all times. In general the RMS error arising from the CP approach is smaller, as a result of the averaging effect of track coalescence and the fact that the y position is the same for both targets.

For the case $S = 4$ the x coordinate RMS behaves similarly as for the case $S = 3$, but including an extra list (more time depth) has a result that less tracks coalesce (due to the best hypothesis being in general stronger than the subsequent), and thus the RMS error in DM increases compared to the case of $S = 3$ due to some track switches that were not captured by the algorithm.

The inital position of the targets is such that a simple velocity gate assures correct data association. Thus the estimate for target 1 is correctly assigned the track ID T_1, and similarly track 2 (T_2) represents target 2. After a certain time, the trajectories get closer (separated c units in the x coordinate and 0 in the y coordinate) and become parallel, as stated before. For the TmHT methods, if at a certain time the track estimates are merged (the tracks coalesce, i.e., are contained in the same $\mathcal{C}$-set, and a single track estimate is kept that contains the two targets) the track IDs are lost. Thus, in case of significant contention between hypotheses, the IDs are no longer available. These IDs will be re-initiated whenever the tracks can be uniquely identified again. On the other hand, the conventional MHT will keep the IDs of the tracks when there is hypothesis contention (the algorithm has no way of discriminating this), the worst case being the occurrence of a track switch. That is, track T_i now represents target j, $j \neq i$, i.e., the IDs represent the wrong targets. The calculation of the RMS errors takes this into account, so that when a single coalesced track is present, the RMS error will be usually less than when track switch occurred as a result of the distance between the true target and the switched tracks being larger than the distance between the true targets and the merged track. In the case of no track switch, the merged estimate yields larger RMS errors.

Table IV shows measures of track switching and track coalescence for the case of a window of $S = 3$ frames (the case $S = 4$ has similar results). Two measures are used for the track coalescence, one is the percentage of tracks that have coalesced at some point during the time span of the experiment, and other is the percentage of time that this coalescence lasted. If there are track switches, the percentage of them occurring is also shown. It can be seen that the TmHT-CP does combine track estimates for all the separations presented, as a result of the weak condition for track combination, and that the tracks remain combined for most of the time span of the experiment. This prevents any track switch, but worsens the estimation RMS, as shown before. On the other hand, the TmHT-DM does combine tracks when those have relevant information in common. In

TABLE IV
Track Coalescence and Switch Measures for the 3 Association Methods, as a Function of Different Separations c (in the x direction), using a Window of $S = 3$ Frames

	TmHT-CP		TmHT-DM			MHT
c	% coalescence	% duration	% coalescence	% duration	% switches	% switches
14	100	94	0	0	1	1
12	100	93	9	4	2	9
10	100	98	42	14	3	20
8	100	99	98	30	4	56

this way tracks that are more likely to be confused, as is the case when their distance diminishes, have a larger percentage of track coalescence. Such coalescence lasts as long as there are association hypotheses that are very similar, thus not spanning the whole experiment time length. Note that some track switches still occur, but the number is significantly smaller when compared to the MHT.

5. CONCLUSIONS

A method has been presented to combine the m most significant hypotheses in a T2TA problem, which gives consistent system track estimates in the case of contention (correlation ambiguity), as opposed to using only the best hypothesis. The use of the m-best hypotheses has two main advantages. The first is the correct calculation of the variances due to mixing of related estimates, which improves the consistency of the estimator. The second is the ability to quantify the difficulty in the association by checking the total probability of the $\mathcal{C}$-sets (the combinable tracks). If this probability is close to one, the estimation for the target represented by the associations in the $\mathcal{C}$-set can be considered reliable. If the total probability is greater than one, the estimate obtained should be considered in a special manner, as it is based on wrong/mixed tracks due to missasociation.

The method proposed here, called Direct Mixture (DM) is somewhat similar to the Coordinated Presentation Mixture (CP) [5], but improves upon it in two aspects: the criterion for $\mathcal{C}$-set inclusion, and the use of likelihood based probabilities for association combination, which have been shown to improve the quality of the resulting estimates.

Overall, the proposed DM method with $K = 3$ (the overlap requirement between combinable hypotheses) yields more accurate estimates in terms of RMS error and more consistent estimates than both the top hypothesis scheme and CP.

The DM method has also been extended to a dynamic target tracking case. It has proven useful when the distance between tracks is such that there is association ambiguity (otherwise MHT, or even simpler methods as PDA suffice). The fact that estimates are merged when they have significant information in common, measured as the number of common measurements, allows for a decrease of the RMS error. This also causes the track IDs to be lost, but has the advantage of avoiding track switchings.

REFERENCES

[1] J. Areta, Y. Bar-Shalom, M. Levedahl and K. Pattipati
Hierarchical track association and fusion for a networked surveillance system.
Journal of Advances in Information Fusion, **1**, 2 (Dec. 2006), 140–157.

[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), 3–14.

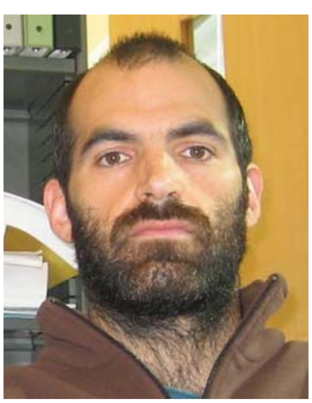
[4] Y. Bar-Shalom, X. R. Li and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.
New York: Wiley, 2001.

[5] I. Bottlick and S. Blackman
Coordinated presentation of multiple hypotheses in multitarget tracking.
In *Proceedings of SPIE Conference on Small Target Tracking*, #1096-1, 1989, 152–157.

[6] S. Gadaleta, S. Herman, S. Miller, F. Obermeyer, B. Slocumb, A. Poore and M. Levedahl
Short-term ambiguity assessment to augment tracking data association information.
In *Proceedings of 8th Intenational Conference on Information Fusion*, Philadelphia, PA, 2005.

[7] K. R. Pattipati, S. Deb, Y. Bar-Shalom and R. B. Washburn
Passive multisensor data association using a new relaxation algorithm.
In Y. Bar-Shalom (Ed.), *Multitarget-Multisensor Tracking: Advanced Applications*, Artech House, 1990, ch. 7. Reprinted by YBS Publishing, 1998.

[8] R. Popp, K. R. Pattipati and Y. Bar-Shalom
An m-best multidimensional data association algorithm for multisensor multitarget tracking.
IEEE Transactions on Aerospace and Electronic Systems, **37**, 1 (Jan. 2001), 22–39.



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 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 at the University of Connecticut. He is also Director of the ESP (Estimation and Signal Processing) Lab.

His current research interests are in estimation theory and target tracking and has published over 370 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate texts *Estimation and Tracking: Principles, Techniques and Software* (Artech House, 1993), *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction* (Wiley, 2001), the advanced graduate 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 government agencies, and originated the series of Multitarget-Multisensor Tracking short courses offered via UCLA Extension, 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 Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–87 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–89 was a member of the Board of Governors of the IEEE CSS. He was a member of the Board of Directors of the International Society of Information Fusion (1999–2004) and served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications in 2004–08.

In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS and has given numerous keynote addresses at major national and international conferences. He is corecipient of the M. Barry Carlton Award for the best paper in the IEEE Transactions on Aerospace and Electronic Systems in 1995 and 2000 and the 1998 University of Connecticut AAUP Excellence Award for Research. In 2002 he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. He is the recipient of the 2008 IEEE Dennis J. Picard Medal for Radar Technologies and Applications.

Krishna R. Pattipati (S'77—M'80—SM'91—F'95) received the B.Tech. degree in electrical engineering with highest honors from the Indian Institute of Technology, Kharagpur, in 1975, and the M.S. and Ph.D. degrees in systems engineering from the University of Connecticut in 1977 and 1980, respectively.

From 1980–86 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 370 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. He 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*. Dr. Pattipati has served as the Editor-in-Chief of the *IEEE Transactions on SMC: Part B—Cybernetics* during 1998–2001, Vice-President for Technical Activities of the IEEE SMC Society (1998–1999), and as Vice-President for Conferences and Meetings of the IEEE SMC Society (2000–2001). He was corecipient of the Andrew P. Sage award for the Best SMC Transactions Paper for 1999, Barry Carlton award for the Best AES Transactions Paper for 2000, the 2002 and 2008 NASA Space Act Awards for "A Comprehensive Toolset for Model-based Health Monitoring and Diagnosis," the 2003 AAUP Research Excellence Award and the 2005 School of Engineering Teaching Excellence Award at the University of Connecticut. He also won the best technical paper awards at the 1985, 1990, 1994, 2002, 2004 and 2005 IEEE AUTOTEST Conferences, and at the 1997 and 2004 Command and Control Conferences.

A Market-based Approach to Sensor Management

VISWANATH AVASARALA
TRACY MULLEN
DAVID HALL

Given the explosion in number and types of sensor nodes, the next generation of sensor management systems must focus on identifying and acquiring *valuable* information from this potential flood of sensor data. Thus an emerging problem is deciding what to produce, where, for whom, and when. Identifying and making tradeoffs involved in information production is a difficult problem that market-based systems can "solve" by allowing user values, or utilities, to drive the selection process. Essentially this transforms the traditional "data driven" approach (in which multiple sensors and information sources are used, with a focus on how to process the collected data) to a user-centered approach in which one or more users treat the information collection and distribution system as a market and vie to acquire goods and services (e.g., information collection, processing resources and network bandwidth). We describe our market-based approach to sensor management, and compare our prototype system to an information-theoretic system in a multi-sensor, multi-user simulation with promising results. This research is motivated in part, by rapid technology advances in network technology and in sensing. These advances allow near universal instrumentation and sensing with worldwide distribution. However while advances in service-oriented architectures and web-based tools have created "the plumbing" for data distribution and access, improvements in optimization of these distributed resources for effective decision making have lagged behind the collection and distribution advances.

Manuscript received August 13, 2007; revised January 6, 2009; released for publication June 3, 2009.

Refereeing of this contribution was handled by Chee Chong and Robert Lynch.

This material is based upon work supported by the Army Research Laboratory under contract number W911NF-04-2-0049 and Tennessee State University under subcontract number 332.77-07.361.

Authors' addresses: V. Avasarala, G. E. Research, Niskayuna, NY 12309, E-mail: (avasaral@research.ge.com]; T. Mullen and D. Hall, College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16802, E-mail: ({tmullen, dhall}@ist.psu.edu).

1557-6418/09/$17.00 © 2009 JAIF

1. INTRODUCTION

With the advent of small, inexpensive, low-power sensor nodes that can provide sensing, data processing, and wireless communication capabilities, sensor networks can potentially generate huge amounts of diverse data. However, just because the data can be produced, does not mean that it should be. Sensor networks are constrained by limits on sensor "attention" that include limitations on battery power, bandwidth, and the number and type of measurements that the sensor can handle at any one time. In addition, sensor networks can include data collection entities that operate on very different timescales, from human reports to high-speed video-frame collection of images. System users (who may be human, software agents or data fusion processes) each have their own individual tasks and priorities, but share a common sensor resource pool. The sensor manager's job is to efficiently allocate sensors to end-user tasks so as to maximize end-user utility while simultaneously minimizing the cost of collecting, storing, and processing the data. Sensor managers must also consider the interplay between various network resources, weighing tradeoffs between resource constraints such as battery power, bandwidth, and sensor accuracy.

For our current work, we assume that end users belong to a common overarching non-commercial institution. Example application areas for such networks include: (1) network-centric warfare, in which multiple sensing platforms, sensor nets, and individual soldiers with sensors interact to allow rapid tactical situation assessment and threat assessment [11, 12], and (2) monitoring of the environment via ground-based, airborne and space-based sensing systems.

In recent years, information-theoretic approaches have emerged as a promising paradigm for the development of a comprehensive sensor management for multi-task, multi-sensor networks. These techniques rely on optimization of a certain information-theoretic measure like cross-entropy [5, 20] or information gain [6, 27]. Kastella [18] used cross-entropy to determine the optimal search order for detection and classification problem. Kolba et al. [20] extended this framework to permit operation with uncertain sensor probabilities. McIntrye et al. [25, 26] used information gain (the entropy change in environment for a given sensor allocation as the predicate for their hierarchical sensor management architecture. A valuable advantage of information-theoretic approaches is that they are highly flexible and can be easily adapted to new problems. However, information-theoretic sensor management is concerned primarily with scheduling the data-collecting entities (sensors) and other network resources such as energy usage and communication bandwidth have to be considered separately. Additionally, information-theoretic sensor management approaches are myopic in nature, since they optimize some measure of the "quantity of information" obtained during a particular round of scheduling and neglect the "value of information" to the mission objectives.

Non-myopic sensor managers need to solve a highly complex multi-period scheduling problem, since most network tasks like target tracking occur over multiple periods of scheduling. Techniques based on approximate dynamic programming have been developed for this problem. In [5], Castañon considered a multi-grid, single sensor detection problem. Under certain assumptions about the target distributions and probability distribution of sensor measurements, Castañon proved that the optimal allocation policy would be to search either of the two most likely target locations during each round of scheduling. In [6], Castañon considered the problem of dynamic scheduling of multi-mode sensor resources for the classification of multiple unknown objects. To solve this problem, the author proposed a hierarchical algorithm based on a combination of approximate dynamic programming and non-differentiable optimization techniques. Washburn et al. [43] formulate a single-sensor, multi-target scheduling problem as a stochastic scheduling problem and use the Gittin's index rule to develop approximate solutions. Williams et al. [44] consider a single-target, multi-sensor allocation problem with communication constraints and use adaptive Lagrangian relaxation to solve the constrained dynamic programming problem. Schedier et al. [37] have used approximate dynamic programming to allocate gimbaled radars for detecting and tracking targets over a multi-horizon time period. The authors use a three-phase rollout algorithm with the following stages a. generation of candidate sensor allocations b. generation of alternate sensor plans based on results from the first component c. evaluation of the alternate sensor plans to calculate an approximation of the reward function. The details of the implementation of these components are not explained in the original paper. However, for the three-sensor simulation that was presented in their paper, simple heuristics to generate feasible solutions and to evaluate solution performance would have sufficed. These approaches for non-myopic sensor management are pioneering, but substantial further research is required to adapt them to the generic multi-sensor, multi-task sensor management problems.

The network-centric environments that we are interested in also must consider issues related to privacy and communication costs. For example, in network-centric warfare applications, multiple distributed entities accomplish different tasks by connecting decision makers, effectors, and information sources to a common network [27]. Therefore, task information may be localized across individual users. Allowing all required task information to be accessed by an optimization routine is communication-intensive and may violate privacy issues in a distributed environment. Pricing mechanisms can be designed to address privacy issues and minimize communication requirements [42]. Also, market-based approaches offer an inherently distributed mechanism that can compare "apples" and "oranges" using the common numeraire of money, thus reducing communication overhead to the single dimension of price. Under certain assumptions, price systems have been proven to provide the minimum dimensionality of messages necessary to determine Pareto-optimal allocations [16].

For the above reasons, we believe markets based on combinatorial auction mechanisms are a promising paradigm for a comprehensive sensor management. A combinatorial auction is an auction based on exchanging item bundles (e.g., sensor readings+channel transmission) rather than single items. In earlier work on distributed multi-agent sensor management, Lesser et al. [22] surmised that combinatorial auctions could be a promising path for market-based sensor allocation. Shortly after our initial work on combinatorial auctions for sensor management [2], Ostwald et al. [32] also published preliminary work on using combinatorial auctions to find optimal sensor settings in a distributed radar array. The authors optimize a domain-specific and myopic utility function using a combinatorial auction mechanism during each round of scheduling. Resource constraints other than the sensor schedules are not considered.

A generic market-oriented approach to sensor management that is customizable for different sensor network scenarios must address several key issues. The first issue is the mismatch between what users want to buy (e.g., tracking and identifying a target with a specified accuracy) and what network resources are offering (e.g., cpu, battery power, bandwidth, and sensor directivity and operation mode). The problem becomes even more complicated when we consider that different combinations of sensors can be used to track a target, but each combination of sensors may give a different quality of service (QoS). To accurately assess and bid for different sensor combinations, users would need to know the operating parameters of each sensor, and to calculate the QoS for various combinations of sensors. This leads to the second issue of *preference elicitation*, or eliciting user valuations for all possible combinations of resources to different tasks/users. Clearly in this setting, preference elicitation can be computationally and/or communication intensive. For example, if there are n sensors and m tasks, $((2^n - 1)m + 1)$ utility valuations must be acquired by the sensor manager from the user to calculate an optimal allocation. The third issue is that *winner determination*, or determining an optimal allocation given all bids, is an NP-hard problem [35]. Although fast algorithms have been developed, thanks in part to ecommerce-driven advances, these algorithms may not always meet real-time requirements.

To address the above issues, we proposed a framework for sensor management using a market-based architecture called MASM (Market-Architecture for Sensor Management) [15, 30, 33]. The sensor manager handles the mismatch between what providers (i.e., sensors) offer and consumers (i.e., end users) want by

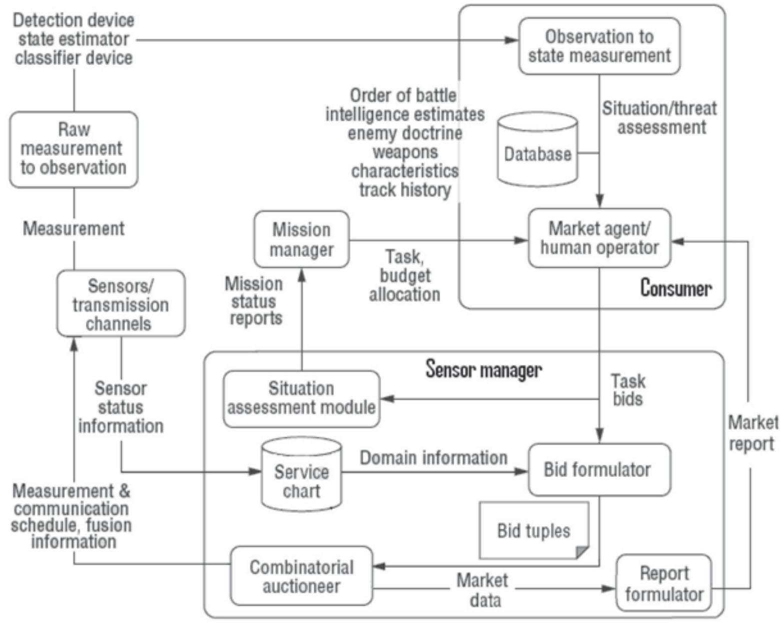


Fig. 1. Single-platform market architecture for sensor management.

providing a mapping between user tasks and network resources. Users bid on high-level tasks, while service mapping components convert the high-level user tasks to low-level sensor tasks and finally to actual bids. Once the necessary bids are created, an auction winner determination algorithm computes the final resource allocation. Both the bid formulation and winner determination steps are computationally expensive. Traditionally, humans have been mainly responsible for the bid formulation step, with computational auctions focusing on the winner determination step. One of our contributions has been to develop an approximate algorithm, called Seeded Genetic Algorithm (SGA) [29], that combines these two steps and achieves polynomial run times with a modest loss of optimality. Our earlier work [2, 3, 28, 29] described a high-level framework for MASM, but did not provide any implementation details. This paper describes the implementation of MASM, including the auction protocol, pricing algorithms for network resources and heuristics for avoiding myopic scheduling behavior.

We currently focus on a single-platform design, although we plan to extend this model to multiple platforms and sensor network environments. While we draw from recent advances in ecommerce-based market research, we describe the significant challenges in adapting this approach to reflect typical sensor management environments. We test our prototype sensor management system using a multi-sensor, multi-user simulation framework that models bandwidth and battery power constraints. Comparisons to a priority-based information-theoretic system show that market-based algorithms hold promise for developing comprehensive sensor management systems.

It should be noted that the present approach has limited applicability to smart dust environments, where the number of sensors could be on the order of few hundred thousands. In these environments, the communication costs of relaying sensor measurements to the sink are the dominant costs of network operation. For these environments, a centralized auctioneer cannot be used because of the communication costs involved. Instead, task utility information and price information should percolate to the node level, where individual nodes decide on what actions to perform. Mainland et al. [24] have proposed a price-based decision system for smart dust environments, and Padhy et al. [33] have proposed a utility-based model.

Our paper is organized as follows. In Section 2, we describe the MASM architecture and provide an illustrative scenario in Section 3. Section 4 talks about our continuous combinatorial auction (CCA) protocol developed to minimize communication involved in market operations. Section 5 describes the pricing mechanisms that have been developed to enforce resource constraints in the market. Section 6 introduces an agent learning scheme for market agents to assist users in formulating optimal bidding parameters for different tasks. Section 7 describes our simulation environment, while Section 8 describes our results. We summarize our findings and discuss future work in the last section.

2. MASM

Our current single platform design for MASM is shown in Fig. 1, and derives from the sensor management architecture proposed by Denton et al. [8]. The mission manager (MM) assesses mission-level decisions (e.g., assigning task priority to a mission goal), allocates tasks and budgets to end-users. Within the mission manager, approaches such as goal lattices (which relate high-level mission goals to lower-level actionable tasks) can be used to measure the criticality of various low-level goals to the overall mission goals, and thus help to determine their respective budgets. Kenneth Hintz and Gregory McIntyre [14] used goal lattices to compute the relative weights of actionable tasks (such as tracking) on the basis of high-level mission goals. Rajani Muraleedharan and her colleagues [30] used goal lattices to determine weights for combining various objectives to optimize routing in a sensor network. Newer developments include dynamic goal lattices [15] that can support more dynamic goal generation from a set of predefined goals.

The sensor manager (SM) acts as a competitive market for buyers and sellers of sensor resources. Sensors and transmission channels are modeled as sellers. Sensors sell their sensor schedule (i.e., their "attention") and transmission channels sell raw bandwidth. End users, or consumers, of the sensor network are interested in higher-end products such as target tracks, environmental searches, and target identification. MASM maps between these high-level tasks and actual resources available in the market using its combined service chart/bid formulator functionality.

MASM provides this functionality in two different modes, either exact service mappings (E-MASM) or approximate service mappings (A-MASM). When the number of sensors is small and the real-time constraints are relaxed, E-MASM mode provides an exact service mapping. In other words, given a task and a set of possible resource combinations that can be used for that task, E-MASM will explicitly calculate the utility of assigning each combination to the task using domain information and task-specific utility functions provided by a service chart. Given n sensors in the network, and m tasks, then in the worst case, $(2^n - 1)m$ bids on resource combinations might have to be formulated. A standard combinatorial auction winner determination algorithm [1] then determines the optimal allocation. One approach used to speed up the bid formulation auctions and the winner determination optimization is to generically restrict the type of bids considered for resource allocation. For example, one could place a bound on the maximum number of items in a bid. Polynomial algorithms for bids with certain special structures [35] are available. However, imposing generic constraints on bid types can lead to market inefficiency. An alternate approach is to use domain-specific knowledge to intelligently restrict the number of resource bids formulated. For example, if the types or locations of sensor resources that can be used to accomplish a particular task are limited, the combinations of resources that need to be considered can be reduced.

When the number of sensors is large and real-time constraints are strict, explicit mappings are no longer feasible, and the A-MASM mode, with approximate service mappings, is used. Instead of the bid formulator explicitly formulating combinatorial bids for each user task, MASM searches the search space of useful sensor combinations directly using a polynomial, anytime evolutionary algorithm [29].

3. SCENARIO EXAMPLE

To illustrate the MASM system, we describe a simple scenario with two users and two sensors. Assume that a particular user is interested in searching and identifying reconnaissance drones approaching from a certain region R_1. Let the task that the user wants to accomplish using the sensor network resources be the reduction of entropy of the probability distribution of the existence of reconnaissance drones in region R_1 to less than a threshold ε_1 (task A).

The user should submit a bid to MASM in the following format:

(type: *search/identify*
entity: *reconnaissance drone x*
region: R_1
quality: ($entropy < \varepsilon_1$)
price: P_A)

where P_A is the user's bid price.

Assume that another user is interested in estimating accurately the position of an already identified slower moving reconnaissance drone. Let the task that this user is trying to accomplish using the sensor network resources be the reduction of the track uncertainty (as measured by some reasonable metric such as state vector covariance error) to less than ε_2. The user should submit a bid to MASM in the following format:

(type: *track*
entity: *reconnaissance drone y*
quality: ($covariance\ error < \varepsilon_2$)
price: P_B)

where P_B is the user's bid price.

Assume that two sensors, a forward looking infrared (FLIR), or infrared camera, and a radar are available to the SM for accomplishing these tasks. Note that any two sensors will generally have different abilities to locate and identify targets depending on characteristics such as environment conditions, target characteristics, and target-sensor geometry. In other words, certain sensors, or combinations of sensors, can provide more or less value for task completion, and thus the bid values for different sets of sensors may also vary. To express this, MASM generates combinatorial bids in exclusive-or format for each user's task as shown in Table I during each of scheduling. The exclusive-or format ensures that task A can win either bid 1 or bid 2, but not both. For each bid in Table I, the bid amount is represented using the format $P_{U,S,t}$ where U is the task identification, S is the given sensor combination number and t indicates the scheduling round. This notation is used to indicate that bid prices depend on sensor combination and user

TABLE I
Bids Generated by MASM for Sample Scenario During the First Round of Scheduling

Task A's XOR bids		Task B's XOR bids	
Bid 1	Bid 2	Bid 3	Bid 4
(type: *search/id*	(type: *search/id*	(type: *track*	(type: *track*
entity: *reconnaissance drone x*	entity: *reconnaissance drone x*	entity: *reconnaissance drone 3*	entity: *reconnaissance drone 3*
region: R_1	region: R_1		
sensors requested: *FLIR*	sensors requested: *FLIR and Radar*	sensors requested: *FLIR*	sensors requested: *FLIR and Radar*
quality: ($entropy < \varepsilon_1$)	quality: ($entropy < \varepsilon_1$)	quality: ($covariance\ error < \varepsilon_2$)	quality: ($covariance\ error < \varepsilon_2$)
price: $P_{A,S_1,1}$)	price: $P_{A,S_2,1}$)	price: $P_{B,S_1,1}$)	price: $P_{B,S_2,1}$)

task. The value of $P_{U,S,t}$ is calculated by the SM using the bid prices of the original consumer bids (see Section 4 for details). Until the tasks are complete, the SM monitors the progress of the tasks and adjusts the bids accordingly.

We describe the methodology used by MASM to generate the bids for resources during each round of scheduling in the next section. The combinatorial auction winner determination algorithm is then used to calculate the optimal resource allocation, given the MASM bids.

4. CCA PROTOCOL

In this section, we describe our continuous combinatorial auction (CCA) protocol. The CCA protocol was designed to increase the computational and communication efficiency of our market-based scheduling algorithm. Since MASM uses discrete time slots to schedule resources, most user tasks, like tracking a target, require acquiring resources over multiple time slots. Each round of scheduling can either occur periodically at fixed times, or randomly. A simplistic allocation of resources across multiple time slots can occur in two ways: i) Users send in a bid that covers resource needs across multiple time slots. The SM updates the schedule upon receiving each new user bid. ii) Users send in a bid for the current time slot only. After each round, users update their requirements based on what was received in the last scheduling round, and send in an updated bid for the next time slot.

The first approach is computationally expensive. Determining the optimal scheduling for n sensors over a time horizon T is exponentially complex in n and T. Clearly, the second approach is communication intensive. We designed the CCA protocol to avoid the communication and computation requirements of using markets for sensor management. Below, we describe the CCA protocol in detail.

CCA executes each of the following steps (except initialization, which is executed once at the start of operations) during each round of scheduling.

4.1. Initialization

Auctioneer initializes the prices for all the resources. It informs the users about the set of tasks that it will accept bids for.

4.2. Update Bids

At the beginning of each round, users can i) send new bids, ii) remove their current bids from the auction, iii) modify the parameters of their existing bids. User bids are of type $\langle t, p\rangle$ where t is the task description, which includes the task type, and final task quality desired by the user and p is the price that the user is willing to pay. For example, the task description for a bid to track a target x such that the trace of the covariance matrix of the target estimate is less than 0.001 is as follows:

(type: *track*
entity: *target x*
quality: (*trace of covariance matrix* < 0.001))

The auctioneer predefines the set of tasks that the user can bid for and the bid format. Here we make the standard assumption that a scalar valued "quality" measure can be calculated using the various task parameters. For example, for target tracking, the trace or the determinant of the covariance matrix can be used as one measure of target track quality [13, 31, 36]. However, under certain circumstances, it is advantageous to use more elaborate task descriptions (see [17] for related discussion). For example, the requirement to identify a target with a given level of specificity and level of confidence may require extensive models of multi-sensor performance in complex observation environments. These are application-specific, and would need to be developed for the particular application being considered.

4.3. Update User Requests

Auctioneer accepts new bids or updates to existing bids during each round of scheduling. If the auctioneer receives no message regarding a particular bid, the bid stays active and competes for resources in the current auction round.

4.4. Resource Bid Formulation

Since user bids are for high-level tasks, the auctioneer needs to compose bids for actual resources from them. This responsibility is handled by the bid-formulator module in E-MASM (explicit formulation of bids for resources is not required in A-MASM). Let the user bid on a high level task T at time t_i with price P_T. A high level task, such as tracking a target to a required accuracy, might require resources over multiple rounds of scheduling. For each time slot t, the auctioneer constructs bids on each resource set, S, that can be allotted to task T. The auctioneer needs to calculate the price associated with resource set S for task T during each round of scheduling, based on the user bid price P_T. To correctly align task priorities with bidding price, we have devised a novel mechanism for resource price determination. For a resource set S, the auctioneer computes the bid price for a resource set as the percentage of the user task completed by the resource set given the current task status. To determine the percentage of task completed by a resource set S, we cast the problem in terms of optimally scheduling sufficient readings from a canonical sensor A to meet the quality of service (QoS) task parameters. Let the task T require on average n_a consecutive schedules of the standard sensor A to be completed (task is considered complete, when the task quality meets the QoS threshold in the task bid). Suppose a resource bundle S is used when the task quality is q, and the expected number of standard sensor readings required is reduced to $\bar{n}_a$ Then $f_{S,T,q}$ or the percentage of the task completed by resource set S when the current task quality is q, is equal to the percentage savings in the required number of canonical sensor readings.

$$f_{S,T,q} = (n_a - \bar{n}_a)/n_a.$$

There is a possibility that $f_{S,T,q}$ is negative ($n_a < \bar{n}_a$). For example, in spite of allocating sensing resources, the inaccuracy associated with a target estimate might increase with time. To avoid negative prices for bundles of resources, the bid price for $P_{S,T,q}$ (bid price for allocating resource set S to task T during a particular round of scheduling when the current task quality is q) is calculated as

$$P_{S,T,q} = P_T * f_{S,T,q} - P_T * f_{\phi,T,q}$$

where ϕ is the null set.

The auctioneer uses this price to prioritize between different tasks during a particular schedule. However, this price is not charged to the user. Users are charged only at the end of a successful task completion, or if they choose to withdraw a bid before the task could be completed by the SM (see the round termination step). Calculation of n_a and $\bar{n}_a$ can be made faster, by storing task specific performance data for the canonical sensor as QoS charts. For example, consider a user bid for searching a particular grid for potential threats, when QoS is measured in terms of entropy. A sample QoS chart is given in Fig. 2, and shows the expected fall in entropy with standard sensor readings. Let a user bid specify a task for reduction of entropy of a particular grid from e_i to e_f. If a resource bundle S is expected to reduce the entropy from e_i to e_j after the next reading, then

$$f_{S,T,e_i} = (n_j - n_i)/(n_f - n_i).$$

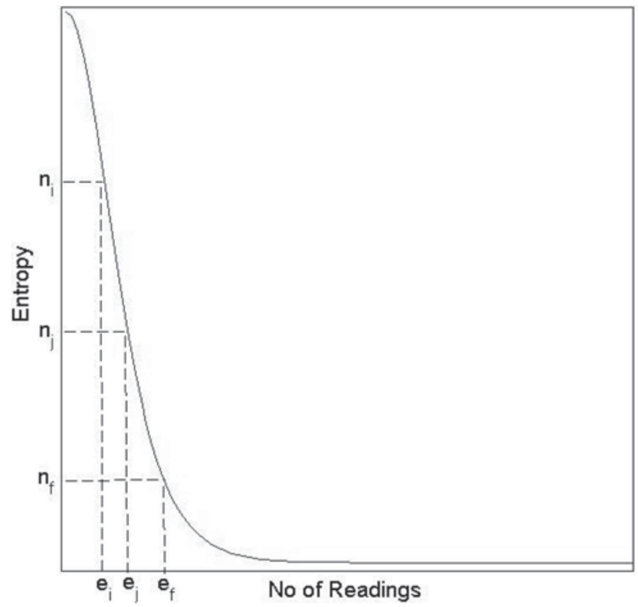


Fig. 2. Illustration of calculation of bid prices for resource bundles using QoS chart.

Creating a QoS chart for a sample task is illustrated in Section 7.

Resource formulation is the slowest link in CCA protocol, but heuristics can speed this step up. For example, for certain tasks, it may be feasible to use only a few kinds of resources. Thus, if a task requires only acoustic data, then only the acoustic radar sensors need to be considered. However, in the worst-case scenario, the number of bids is exponential in the number of sensors. This is clearly infeasible in case of large systems and thus E-MASM is not scalable to large systems, limiting its effectiveness. The A-MASM formulation avoids explicit bid formulation, and hence maintains polynomial run-times, both in number of resources and users by using our SGA algorithm, an approximate polynomial-time algorithm. For a detailed description of this algorithm, and a comparison of A-MASM and E-MASM time performance, see [34].

4.5. Resource Allocation

Resource bids, obtained from step 3 are exclusive-OR bids in the form $\langle S_1, P_1\rangle xor \langle S_2, P_2\rangle \ldots xor \langle S_n, P_n\rangle$. This bid indicates that during the current round of scheduling, the user is willing to pay a price P_1 for the resource bundle S_1 and a price P_2 for S_2, but only the maximum of P_1 and P_2 for $S_1 \cup S_2$. The auctioneer needs to translate these bids to OR bids, so that standard integer programming formulations [1] can be used. An OR bid of the form $\langle S_1, P_1\rangle or \langle S_2, P_2\rangle \ldots or \langle S_n, P_n\rangle$ indicates that the user is willing to pay $P_1 + P_2$ for the bundle $S_1 \cup S_2$. This can be done by the addition of phantom items [9]. The idea is to translate an exclusive-OR bid B-xor of the form $\langle S_1, P_1\rangle xor \langle S_2, P_2\rangle \ldots xor \langle S_n, P_n\rangle$ into a B-OR bid of the form $\langle S_{1b} \cup b, P_1\rangle or \langle S_2 \cup b, P_2\rangle \ldots or \langle S_n \cup b, P_n\rangle$, where b is a phantom item. The phantom item b ensures that a maximum of only one bid from the OR bids can be labeled as winner (since each item can be allocated to maximum of one bids). Once the bids are translated into the "OR" format, the winner determination problem becomes a standard integer programming (IP) problem, that can be characterized as

$$\max \sum_{j=1}^{n} p_j x_j \ s.t. \sum_{j \mid i \in S_j} x_j \leq 1 \qquad \forall \quad i \in \{1 \ldots m\}$$

where x_j is 1 if the bid is accepted in the final allocation and 0 otherwise. The IP problem can be solved using a commercial software package like CPLEX. The winner determination problem is NP-hard [18] and in theory, this resource allocation step could prove computationally expensive for E-MASM. Performance of the IP formulation depends greatly on the characteristics of the probability distribution from which bids are generated. For example, the time taken by a problem with one thousand bids and hundred items on a 2.8 GHz Pentium IV processor varied between 0.001 seconds to 5000 seconds, depending on the bid distribution. However, we found that the problems generated by the sensor network simulation are relatively easy for CPLEX (see Section 4).

4.6. Round Termination

The auctioneer updates the costs of resources expended on a particular bid by adding the price of its allocated bundle. For each user bid b, the cost of resources allocated to the bid is updated as

$$C_b = C_b + \vartheta_{s_i}^t$$

where S_i is the bundle allocated to b during the tth round of scheduling and $\vartheta_{s_i}^t$ is the price of the bundle S_i during the tth round of scheduling. This is calculated as the sum of the prices of the individual resources comprising S_i (see Section 5 for explanation of resource pricing).

Also, the auctioneer verifies if the task quality required by each user bid was achieved. When a task is complete, the bids for that task are removed from the auction and the corresponding user is sent the completed task details. The bidding user is charged the minimum of his bid price P_b or the cost of resources spent on the task by SM, C_b.

$$payment_b = \min(C_b, P_b)$$

where $payment_b$ is the fee charged to the user, P_b is the bid price, and C_b is the total cost of the resources allocated to the bid. This fee structure ensures that no user is charged more than their bid price for any task. When $C_b < P_b$, the user has a positive surplus of $P_b - C_b$. An alternate fee structure that divides the surplus between the SM and user is as follows:

$$payment_b = \min(C_b, P_b) + H(P_b - C_b).*\gamma * (P_b - C_b)$$

where $H(x)$ is the Heaviside step function and γ is the percentage of surplus given to SM. If the user withdraws a bid before the task demanded in the bid is completed, the SM charges the user

$$payment_b = \min(C_b, f_{,b*} P_b) + H(f_{,b,*} P_b - C_b). \\ *\gamma * (f_{,b} * P_b - C_b)$$

where $f_{,b,}$ is the percentage of the task that is already completed by the SM. This is calculated using QoS charts (as described in the resource bid formulation step). This fee structure has been designed to mitigate the impact of dishonest user behavior (see Section 8 for details).

Finally, the auctioneer updates the prices of the resources based on the demand in the current round. We describe how prices are updated in the next section. The auctioneer then updates the resources about their schedules during the current round and sleeps until the next round of scheduling begins.

5. PRICING MECHANISMS

To set prices for individual resources, we use a pricing protocol similar to the tatonnement process. *Tatonnement* is an iterative procedure for finding equilibrium prices based on the search parameter (e.g., price or quantity) [21, 34, 40]. The price adjustment process starts with an auctioneer communicating an arbitrary price set to the users. The users compute their demand for the first good at the given prices and communicate it to the auctioneer. Depending on whether the aggregate demand for the first good is positive or negative, the auctioneer either increases or decreases its price. This process continues until a price at which aggregate demand for the first good equals zero is reached. This process is then repeated for the second good and so on. At the end of the first cycle, only the last good is guaranteed to have a zero demand, but assuming gross substitutability (i.e., when the price of good j goes up, there is a positive increase in the demand for every other good by each user) the price set arrived at after each cycle is closer to equilibrium than the previous one. More refined algorithms using partial derivatives of the demand functions have been developed to search for equilibrium in parallel [38, 45]. Though the gross substitutability assumption is often violated (as in sensor networks), the tatonnement process has been found to give satisfactory results [7].

To use the tatonnement process in MASM, we model the supply and demand functions for a resource at a particular price. MASM estimates these functions using the current resource usage rate. Prices for individual resources are initialized to zero during the sensor network initialization. After each round of scheduling, the prices (ϑ_S^{t+1}) for the resource S for the next round of scheduling are calculated based on the current usage rate of the resource (r_S^t) and the available usage rate of a_S^t.

$$\vartheta_S^{t+1} = \max(0, \vartheta_S^t + \tau * (r_S^t - a_S^t))$$
$$\vartheta_S^0 = 0$$

where τ is the constant which determines the rate at which prices are updated.

The definition of r_S^t and a_S^t is dependant on the resource being modeled. For example, for sensors, we have used the available battery power. Let sensor A be endowed with initial battery power b_i and assume that Sensor A needs to be available for a total operating time of T. At time t, if the available battery power is b_t, then

$$r_A^t = (b_i - b_t)/t \qquad \text{if} \quad t > 0, \quad 0 \text{ otherwise;}$$
$$a_A^t = (b_t)/(T - t) \qquad \text{if} \quad t < T, \quad 0 \text{ otherwise.}$$

Ideally, the tatonnement process would update the price of one resource, run the winner determination algorithm to find the new demand for resources, then conduct price updates for the second resource, and so on. However, because of time and communication constraints, all the price updates are conducted simultaneously using the current rate of utilization, during every round. We expect that the results between the two approaches will not be very different, since the usage rates are moving averages and do not vary significantly based on the usage during the current time slot.

6. AGENT LEARNING

In MASM, the SM accepts bids only on a set of pre-defined tasks. The user agent is responsible for decomposing the high level tasks or goals that it has a utility for into a sequence of SM acceptable subtasks on which it can bid. Also, the user agent has to assign appropriate priorities or bid prices to these sub-tasks, so that its overall performance is optimized. Appropriate assignment of priorities to these sub-tasks has a significant impact on agent performance in the market. As an initial exercise, we experimented with agent learning that uses a simple, greedy Widrow-Hoff based learning to optimize bid parameters based on current market data. For reasons of brevity, the details of this approach are not provided here, but readers are directed to [42]. A more rigorous learning method will be the subject of future research.

7. SIMULATION ENVIRONMENT

A simulation environment consisting of a two-dimensional search area involving multiple targets, multi-user and multiple sensors was developed for testing MASM and comparing its performance with other sensor management approaches. The design of the sensor network, including the communication channel, is inspired by the DARPA sensor network implemented to carry out research in sensor management domain [22]. These sensor networks are more platform-based and differ from the emerging field of "smart dust," where the sensor network could consist of millions of sensors.

Our simulation environment is representative of the types of sensors, communications resources, and mission objectives for a tactical military environment. The various sensor parameters we used are based on realistic sensor models and are obtained from [26]. While this is a basic scenario, with a limited number of sensors and targets, it is representative of the types of non-commensurate sensors that would be available for other applications such as environmental surveillance and crisis management systems (e.g., for homeland security). We make a few simplifying assumptions about sensor models since our main purpose is to test SM performance rather than absolute fidelity to field conditions. Below we describe our simulation model.

7.1. Users

Users consist of a set of software market agents that search for and destroy targets. These agents have the ability to attack any position within a range of r meters and any target that falls within γ meters of an attacked position is destroyed. The agents are not provided with any sensing resources and they depend on the sensor network for obtaining information about the environment. They bid for sensor resources during each round of scheduling and update their status based on information provided by the sensor manager. Initially, agents move along the simulation area with constant velocity v_c, searching for targets. They use the sensor network's resource to search for potential targets and if the probability of target existence within their range exceeds a threshold $p_{\text{threshold}}$, initialize target tracks. Once a target track is initialized, agents can attack a target if the 99% confidence interval of the target's position is less than γ meters. Hence, they are required to track the target to the required accuracy before attacking it. This is again accomplished by buying sensing resources from the sensor network. Agents are assumed to have a utility u_t for destroying a target. To divide the overall utility into utilities for search and track tasks, agents initially use equal priorities. During the simulation run, agents update the search to track budget ratio using the learning method, mentioned in Section 6.

7.2. Targets

Targets are randomly distributed throughout the search area. They move randomly along the city roads with constant velocity v_t, corrupted by a Gaussian white

TABLE II
Sensor Characteristics used in Simulation

Sensor No.	Type	Range	Bearing	Axis	P_D	P_{FA}
1	Doppler	90 m $\pm$ 10%	$1^\circ = 6\sigma$	x	0.95	0.001
2	Radar	30 m $\pm$ 10%	$0.1^\circ = 6\sigma$	x	0.95	0.001
3	FLIR	NA	$0.1^\circ = 6\sigma$	y	0.99	0.001
4	ESM	NA	$1^\circ = 6\sigma$	x	0.5	0.01
5	IR	NA	100 μrad $= 6\sigma$	x	0.99	0.01
6	Radar	30 m $\pm$ 10%	$0.1^\circ = 6\sigma$	y	0.95	0.001
7	Doppler	90 m $\pm$ 10%	$1^\circ = 6\sigma$	y	0.95	0.001
8	Radar	30 m $\pm$ 10%	$0.1^\circ = 6\sigma$	y	0.95	0.001

noise with variance Q. Two different types of targets are modeled (T_1 and T_2). Users have greater utility for destroying T_2 targets. Only T_1 targets were used in the simulation experiments, unless otherwise specified.

7.3. Sensors

The simulation models several different kinds of sensors, including sensors that provide range and bearing, bearings-only sensors, and Electronic Support Measure (ESM) sensors. Measurements of two bearings only sensors, which are not located at the same position, can be combined to create both range and bearing estimates and can be used as a pseudo-sensor. A formal way of modeling sensors is to model their properties, such as bandwidth, wavelength, duration of waveform, signal power per pulse, receiver noise strength diameter of radar aperture. A much simpler modeling technique, in which a sensor's characteristics are characterized by three parameters, its probability of detection P_D, probability of false alarm P_{FA} and bearing [6], is used in this simulation. The simulation environment has eight different sensors of five different types that are located on two different platforms orthogonal to each other. The operating characters of the various sensors are given in Table II. Both the platforms are 100 km away from the search area. Since the distance of the sensors from the simulation area is large, small angle approximation $s = r * d\theta$, where s is the length of area that falls under the sensor's beamwidth, $d\theta$ is a beamwidth of the sensor in radians and r is the distance of the sensor platform to the city (100 km). For a detailed description of the sensor modeling techniques adopted in the simulation, refer to [25, 26]. Each sensor has a battery with e_{initial} units of energy. For the purpose of brevity, all the sensor tasks are assumed to cost zero energy, except the task of transmitting messages. The energy spent in transmitting a message of m bytes over a distance of d meters is calculated as $\dot{\alpha} d^2 m$ where $\dot{\alpha}$ is a constant (see Table III).

7.4. Communication Channel

For communication purposes, a RF communication channel with capacity C is used. All the messages are assumed to be of uniform size M bytes. The communication protocol used is a contention-based protocol (like CSMA/CD) where each agent with a message to communicate senses to see if the channel is busy and transmits if it is not. If two entities start transmitting at the same time, they back off and wait for a random amount of time. The time taken for communication, via this channel, for a fixed number of messages is stochastic. SM enforces the bandwidth constraint by restricting the probability that the time taken for communication is greater than t_{com} to less than $\beta\%$.

7.5. Sensor Manager (SM)

Since the number of sensors is not large, E-MASM formulation is used. Bids on two types of tasks, search and track, are accepted by SM. The QoS for search tasks is in terms of entropy and for the tracking tasks, the norm of the estimate covariance is used. To create the QoS mapping shown in Fig. 2 for the detection task, the following procedure is used. Let the initial probability of target presence in a particular cell be $\pi_0 = 0.5$. (with $\pi_0 = 0.5$, the cell has the highest possible entropy). The initial entropy of the grid H_o is calculated as

$$H_o = g(\pi_0) \qquad \text{where} \quad g(\pi) \text{ is defined as}$$
$$-\pi * \log(\pi) - (1-\pi) * \log(1-\pi).$$

Assume that the canonical sensor A with probability of detection λ_D and λ_{FA} is used for verifying the presence of the target in this cell. Assume that a target is present in the cell. Then, the estimated probability π_n^t of target presence in the cell after n consecutive readings of A, can be calculated using Bayesian analysis. Similarly, let the estimated probability after n consecutive readings by A, if the target is not present in the cell be π_n^{nt}. The expected entropy of the cell after n consecutive readings of A is

$$\bar{H}_n = \pi_0 * g(\pi_n^t) + (1-\pi_0) * g(\pi_n^{nt}).$$

The plot of $\bar{H}_n$ vs. n is used as the QoS mapping for the detection task.

7.6. Information-Theoretic Sensor Manager (ITSM)

To compare the performance of MASM, we needed an alternate sensor manager that can handle multiple

TABLE III
Parameter Values used in Simulation

Parameter	Value	Description
Total no of time slots	500	Total number of resource allocation schedules
n_c	5	No of consumers
n_t	10	No of targets
n_{t_2}	2	No of targets with offensive capabilities
v_c	50 mps	Velocity of consumers
$p_{\text{threshold}}$	0.99	Detection threshold
v_t	50 mps	Velocity of targets
Q	0.01	Variance of Gaussian white noise of target motion
r	50 m	Maximum distance that consumers can attack
γ	1.5 m	Radius of destruction around attacked position
ρ	99%	Required confidence interval length of target's position estimate
u_t	1.0 m	Utility for destroying a target
τ	0.005	Tatonement factor
C	2 Mbps	Bandwidth of communication channel
M	1 Kb	Size of communication message
t_{com}	2 millisec	Maximum time allowed for communication
β	0.01	Required probability that time taken for communication is greater than tcom
α	1 pJ/bit/m2	Energy required to send messages per unit distance per unit message size
e_i	2.5 KJ	Initial energy of sensor batteries

heterogeneous tasks and multiple heterogeneous sensors. As explained in Section 1, the currently available approximate dynamic programming based approaches were not directly applicable to this problem without substantial additional work. For this purpose, we implemented an information theoretic sensor manager (ITSM). Hint and McIntrye [25] used information gain (the entropy change in the environment for a given sensor allocation) as the predicate for their hierarchical sensor management architecture. The amount of information gained can be measured by the change in entropy prior to and preceding a sensor measurement. ITSM calculates the information gain, associated with each possible allocation and schedules the resources as per the allocation with the highest information gain. To ensure that ITSM considers the "value of information," we optimized a weighted measure of information gain, instead of relying on the raw information gain. We used the formulation in Kalandros et al. [17] for priority based information-theoretic based sensor management. Instead of multiplying the information gain by the corresponding task weight, the authors use the formula $I'_{s,t} = I_{s,t} + \log(\theta^t_t)$ where $I'_{s,t}$ is the weighted information gain, $I_{s,t}$ is the information gain obtained from allocating sensor suite s to task t and θ^t_t is the priority of task t. A key issue in the use of ITSM is the priorities that need to be assigned to the various tasks. We exhaustively tested the performance of ITSM by varying the track and search budget ratios and found that the optimal user performance was obtained when a track to search budget ratio of $0.9:1$ was used. To enforce the bandwidth constraint, ITSM does not consider allocations that require bandwidth, which has more than 0.01% chance of crossing the t_{com} limit. The expected time taken for a particular bandwidth consumption was determined by using monte-carlo simulations. ITSM does not model energy constraints and these are handled in the experimental setup as explained in the results section.

8. RESULTS

8.1. ITSM vs. MASM Experiments

Information-theoretic sensor managers schedule sensors to minimize the entropy of the environment. However, incorporating battery power constraints into ITSM is not straightforward since most systems either use adhoc metrics or else do not address power constraints explicitly. Instead of initially testing against multiple adhoc solutions, we compare the ITSM system using two sets of experiments: 1) all the energy requirements of the sensor network are assumed to be zero, and 2) ITSM and MASM consume the same amount of energy for different tasks (as shown in Table III), but ITSM does not use any explicit policy for allocating battery power across the mission. Once a sensor has exhausted its battery, it is not considered in future allocations. In the simulation, the user's primary goal is to destroy as many targets as possible. Therefore, we evaluate sensor management performance by calculating the average number of targets destroyed by ITSM and MASM, as shown in Fig. 3. The left bar graph shows experiments where energy constraints are zero, while the right bar graph shows experimental results when energy constraints are enforced. In both cases, MASM was more successful in meeting user objectives (i.e., in destroying the targets) than ITSM. However, in the second set of experiments, some of MASM's success can be attributed to a better energy enforcement policy and it is not clear from these experiments whether MASM will outperform ITSM for

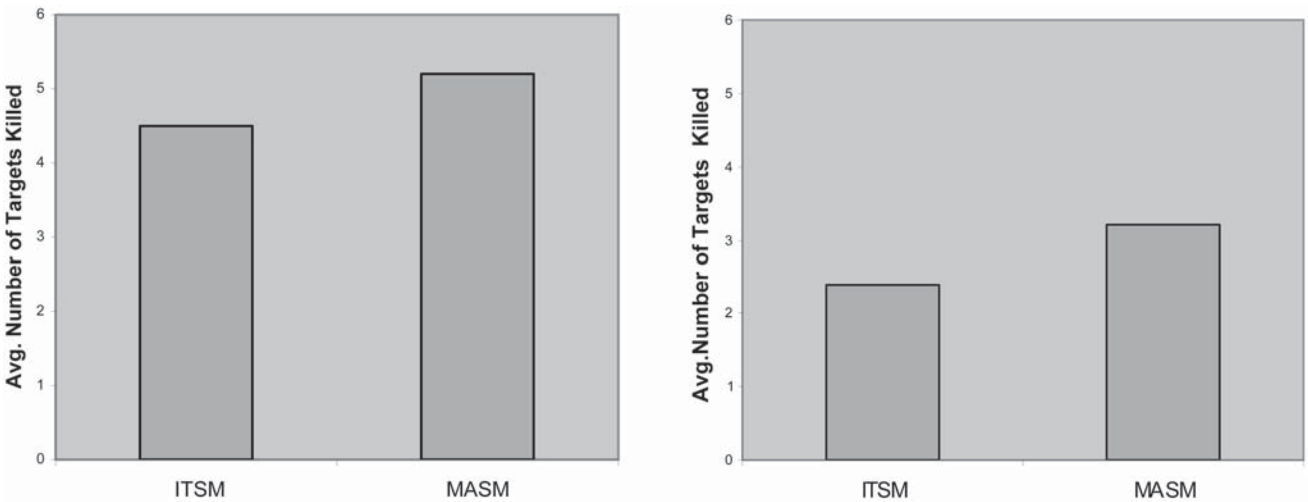


Fig. 3. Comparison of MASM with ITSM (averaged over 100 runs). Left bar graph shows experiments where energy constraints are neglected. Right bar graph shows experiment results when energy constraints are enforced.

any given energy usage policy. We note however, given that MASM does achieve a higher number of targets killed even when the sensor network battery power is "free" for both systems, that this would appear to indicate that MASM's superior performance is not entirely due to a better energy enforcement policy.

Two reasons that MASM outperforms ITSM in meeting user objectives may be that MASM 1) operates to maximize user utility rather than to maximize the information content, and 2) uses prices to prioritize tasks. We discuss these two reasons below.

1) *MASM acts to allocate resources to maximize user utility (as indicated by their bid prices).* Since a user's utility depends on how well the allocated resource set contributes to the user's goals, market-based resource allocation automatically takes goal-related parameters directly into consideration. On the other hand, ITSM concentrates on maximizing information content, neglecting the value of the information to the goals. The priority-based ITSM does a better job than the standard ITSM at incorporating user goals (as priorities) but the system itself has no means of considering a task's progress toward the goal. As an example of why tracking progress toward a goal can be useful, consider the following simple scenario. A single sensor is used to track two targets T_1 and T_2 with equal priority simultaneously. For the first reading, ITSM gets the most information content from tracking T_1, then for the second reading, ITSM gets the most information from tracking T_2. When the confidence interval necessary to attack these two targets is tight, ITSM will never get enough sequential readings to lower the uncertainty sufficiently and will oscillate between the two targets. On the other hand, MASM has equal likelihood for choosing either of the two targets, in any round of scheduling. This happens because the fraction of task completed per reading for either of the target tracks remains constant. Therefore, MASM finishes the tasks in a finite time.

To ensure that our intuition about ITSM vs. MASM was accurate, we implemented the following simple experiment, based on the above scenario, where a single sensor tracks the two targets T_1 and T_2 simultaneously. Target motion is simulated by the equation:

$$x_T(t+1) = x_T(t) + w_T$$

where $x_T(t)$ is the target position at time t and w_T is white Gaussian noise with constant covariance $Q = 0.005$. Targets can be attacked and destroyed if the 99% confidence interval of their position is less than $\beta_{\text{threshold}} = 0.5$ unit. The sensor makes one measurement during each time period, and the measurement equation is:

$$z(t) = x(t) + v(t),$$

where $x(t)$ is the state vector, and $v(t)$ is zero mean white noise with constant variance, $R = 0.03$. Let the initial uncertainties in the position of T_1 and T_2, β_1 and β_2 are equal to 1 unit.

Two sensor-scheduling approaches were implemented. The first approach schedules the sensor to maximize the information gain from the sensor measurement. The second approach schedules the sensor to maximize the utility of measurements, which is defined as the inverse of the total number of sensor measurements required for bringing the targets to threshold uncertainty. The optimal measurement is determined using exhaustive enumeration techniques. The change in uncertainty of the two approaches is shown in Fig. 4 and Fig. 5. ITSM oscillates between the two targets without collecting enough information on any one target long enough to successfully destroy either. The above experiments give an unfair advantage to the utility-based approach since the optimization routine considers multiple sensor schedules simultaneously. In spite of this bias, these experiments offer an insight into the handicap suffered by ITSM due to its inability to take goal related parameters, like $\beta_{\text{threshold}}$, and utility-based calculations

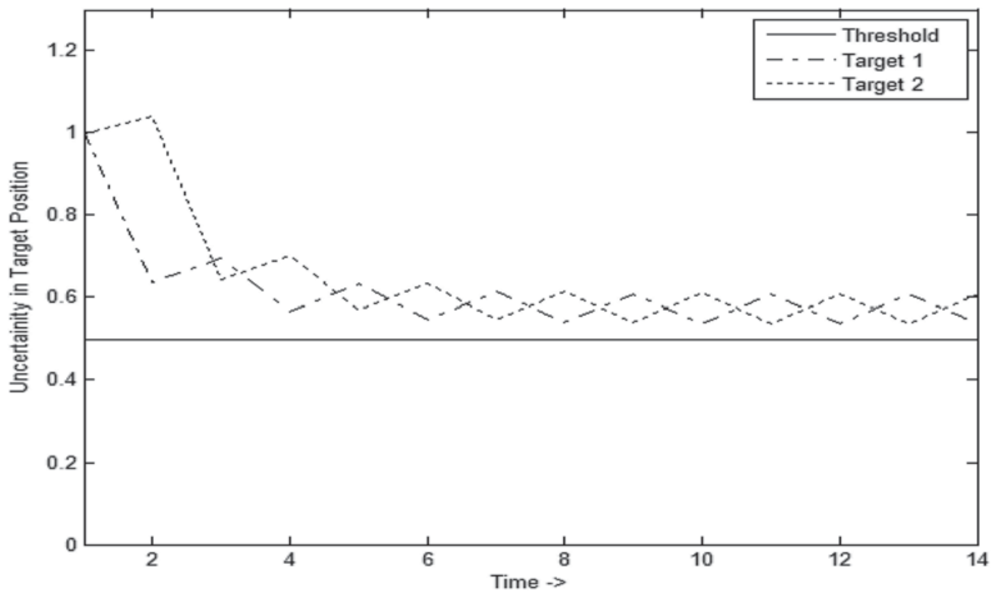


Fig. 4. Change in uncertainty of target tracks, while using an information-theoretic approach.

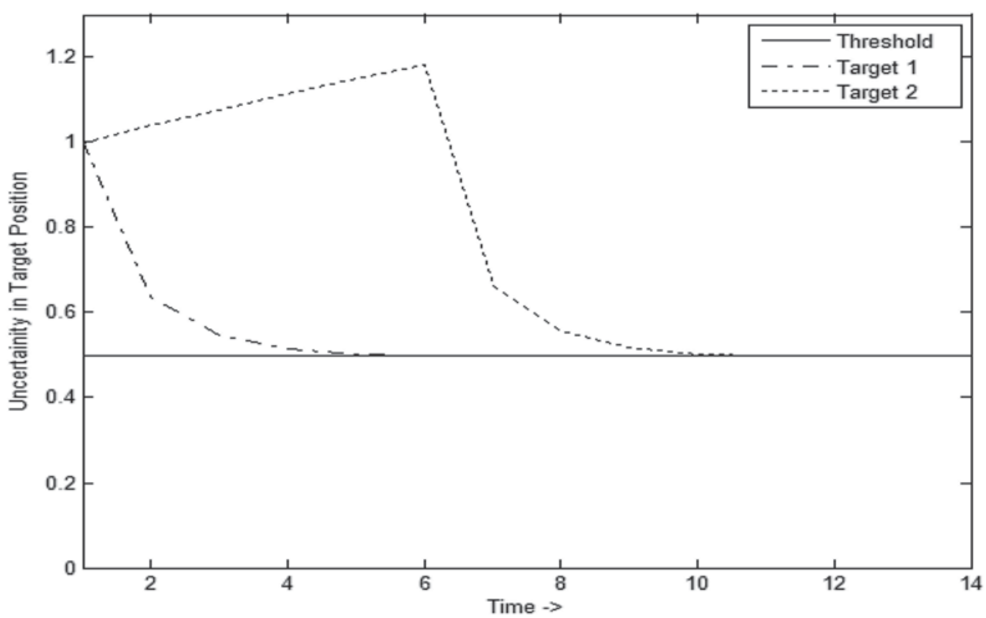


Fig. 5. Change in uncertainty of target tracks, while using a utility-based approach.

directly into consideration. On the other hand, markets provide a principled way to take the utility of a given allocation to high-level goals directly into consideration during scheduling. These results are analogous to those obtained by Castañon [5]. Operating under some assumptions, Castañon considered the problem of determining the optimal sequence of measurements of a single sensor such that the probability that at-least one target is successfully located in a multi-cell grid is maximized. Results demonstrated that the greedy approximation to the optimal solution performed vastly better than an algorithm based on entropy minimization.

2) *MASM prioritizes using prices.* Another advantage of MASM may be due to its use of prices to prioritize tasks while ITSM schedules sensors so as to optimize the information gain from the environment. Although both ITSM and MASM used the same weights to prioritize between tasks in the environment, price-based task prioritization has some inherent advantages. This is because a price-oriented approach has the ability to implicitly reserve resources for future use by high priority tasks, even if no high priority tasks are currently in progress. For example, consider a situation where the first user is tracking a target and the rest of the users are in search mode. Both MASM and ITSM give highest priority to the track task. The first user has a high-budget for a track-bid and bids accordingly. However, during the tracking task, the prices associated with the sensing resources increases since the rate of their battery power usage during tracking is high (refer

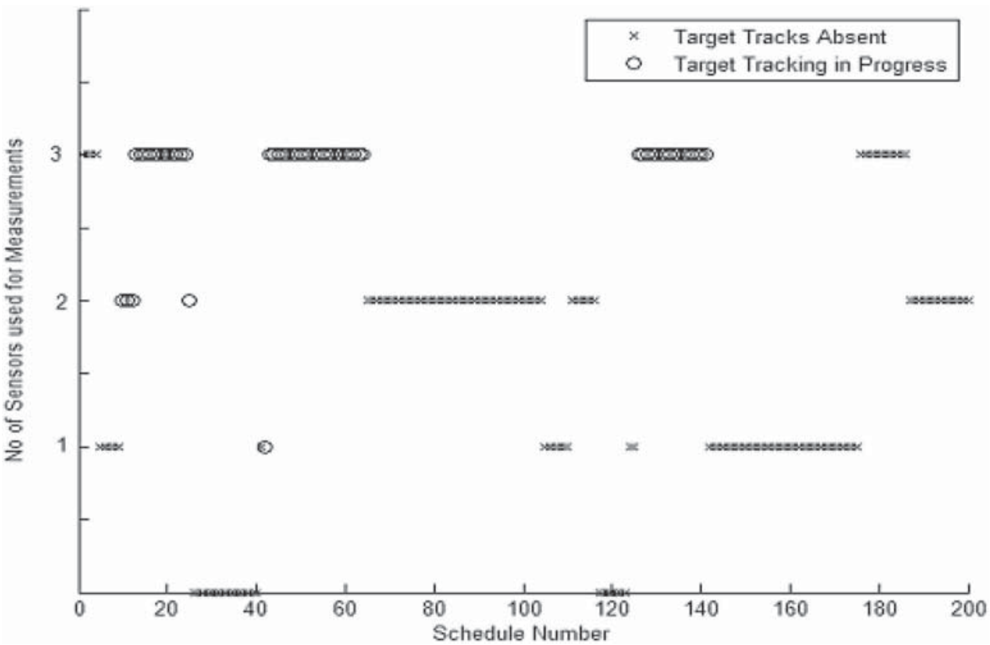


Fig. 6. A comparison of the number of sensors used for measurements, based on whether target tracks are currently in progress or not.

to Section 5 for a description of how prices vary with rate of utilization). After the tracking task is completed and only when detection tasks are in progress, prices of the sensor schedules would have increased. Consequently, sensors will be used at a slower rate during the detection phase, effectively reserving sensors for future higher-priority tasks. However, ITSM has no method of prioritizing between two tasks, except when both the tasks are currently in progress. Fig. 6 shows the number of sensors used during different rounds of scheduling using MASM, where the number of sensors used when tracking tasks are in progress is higher than the number of sensors used when only detection tasks are in progress. When only detection tasks are present, a significant percent of sensors are resting, thereby preserving their battery power for future use.

8.2. MASM-Specific Performance Measures

In addition to our comparison with ITSM, we discuss other MASM-specific performance measures, namely managing resource constraints, task deadlines, scalability, and surplus sharing. The purpose of these experiments is to show how the various "knobs" of a market-based approach can be adjusted to affect performance.

8.2.1. Resource Usage

As explained in Section 5, MASM uses a tatonnement process for enforcing resource constraints such as battery power constraints' using current and available rates of utilization. Fig. 7 shows the price variations of the first three batteries using a tatonnement rate $\tau = 0.005$. Figs. 8 and 9 show the fall in the energy of the first three sensors' battery with successive schedules with $\tau = 0.005$ and $\tau = 0$ respectively. It is clear that tatonnement process is successful in ensuring uniform usage of sensors and in keeping them functional till the end of network operation. For bandwidth also, a similar procedure is adopted. The supply of capacity is constant and is proportional to t_{com}. During round $t+1$, if the price of channel is p_t units/sec and bandwidth consumption is w, then the bound of the $1-\dot{\beta}$ one-sided upper confidence interval of the expected time taken to communicate, η, is calculated based on monte-carlo simulations. The demand at p_t is proportional to η. Values of η for different bandwidth usages are calculated at the start of network operations and stored. The price of a channel during the current round is

$$p_{t+1} = p_t + \tau(\eta - t_{\text{com}}).$$

The price updates for process works as a soft constraint on channel capacity. That is, if the channel is too congested, then prices of the channel will increase till demand for channel capacity decreases. However, it is possible that during certain schedules, the actual time taken for communication is more than the prescribed limit. Figs. 10 and 11 show the time taken for communication for two sample simulation runs with $\dot{\tau} = 0.005$ and $\tau = 0$ respectively.

For the run in Fig. 10, the number of time slots when time taken to communicate crossed the specified threshold is 5. This is within the 0.01% tolerance limit specified by the SM. For the run in Fig. 11, the number of time slots where time taken to communicate crossed the specified threshold is 124. After the 250th schedule, all batteries are exhausted, and the time taken to communicate drops to zero.

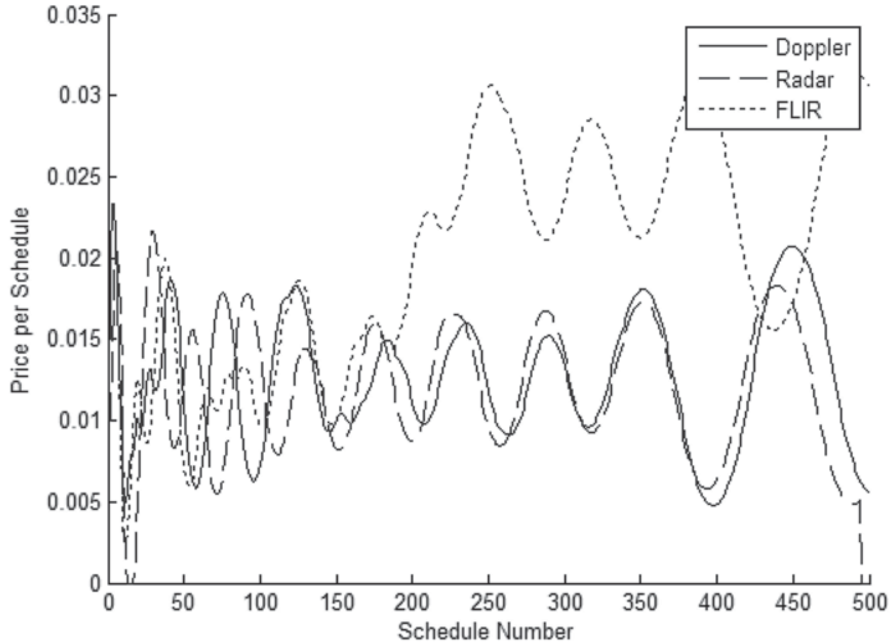


Fig. 7. Price variation of the first three sensors with schedule number for a sample run with tatonnement $\dot{\tau} = 0.005$.

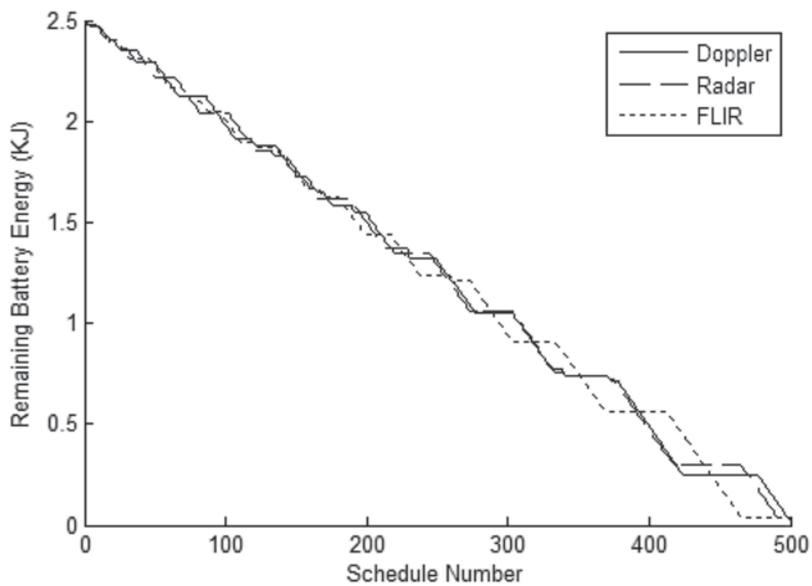


Fig. 8. Energy usage for the first three sensors for a sample run with tatonement $\dot{\tau} = 0.0059$.

8.2.2. Task Deadlines

For some high-value targets, users have a strict deadline to destroy targets after initiating target tracks within t_{kill} schedules. We conducted experiments where a certain fraction of the targets are high-valued. We implemented a user policy of increasing track bids by a factor k, if the detected targets are high-value. For our initial experiments, we used a constant value of 3. However, in the future, optimal k values could be calculated by using the market data. Users recorded an average track time of 7.1 schedules for T_2 versus an overall average of 15.3 schedules, showing how markets can be used to enforce task deadlines.

8.2.3. Scalability Analysis

For the current simulation environment, the IP-based E-MASM formulation can optimally solve problem sizes of up to 10,000 resource bids within a threshold of 10 cpu-seconds on 2.3 GHz PIV processor. Using the SGA-based A-MASM approach, problem sizes of up to 50,000 resource bids can be solved to more than 98% optimality under the same conditions. Larger networks can be accommodated by the approximate technique by compromising on the final optimality. This capability indicates that MASM can be used for fairly large networks, since spatial restrictions often mean that even if a sensor network has thousands of sensors, only a few can be used for a given task at a time.

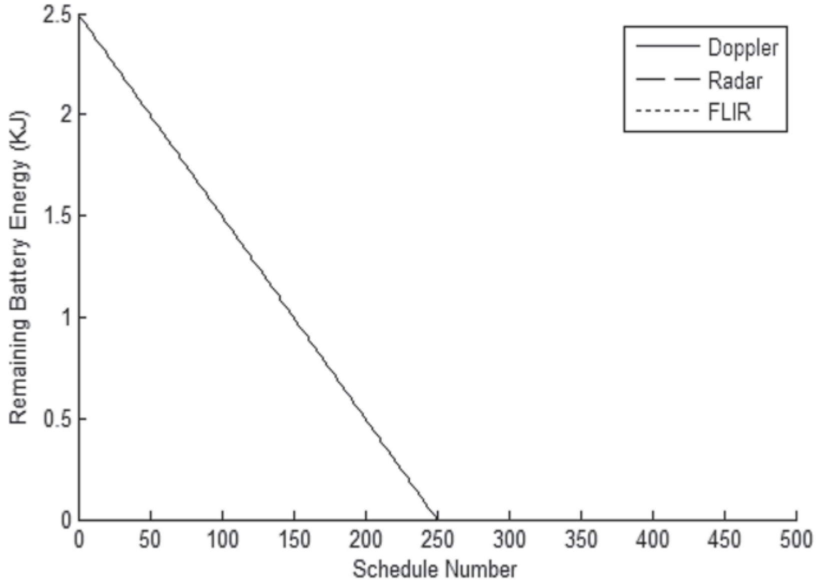


Fig. 9. Energy usage for the first three sensors for a sample run with tatonement $\dot{\tau} = 0$.

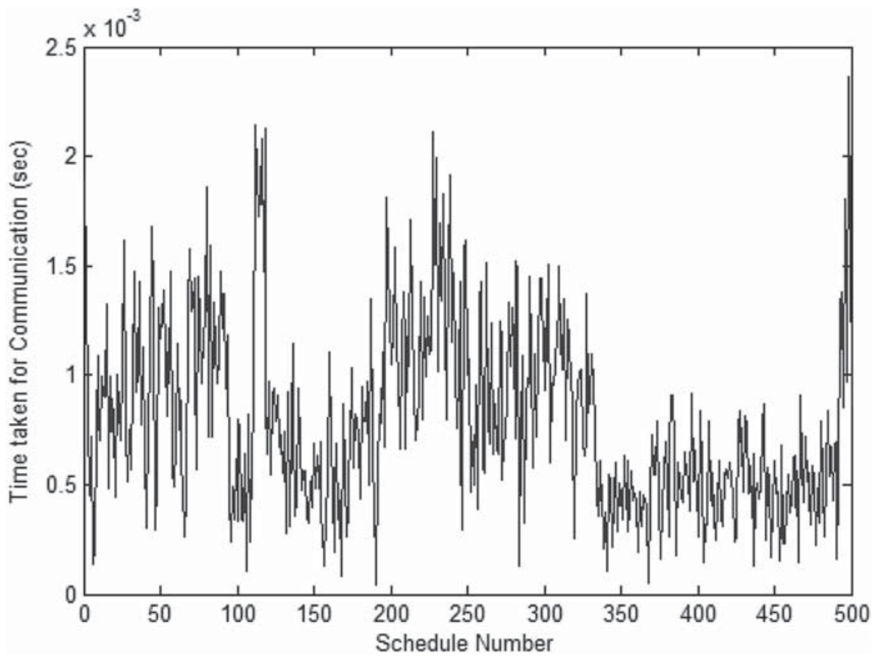


Fig. 10. Time taken for communication for a sample run vs. schedule number for a sample run with tatonnement $\dot{\tau} = 0.005$.

8.2.4. Surplus Sharing

It is possible that sensor networks could have users in a non-cooperative environment, where each agent has an interest only in maximizing its own utility. For example, two different organizations could be sharing the same sensor network resources. Such scenarios require an incentive compatible auction methodology, to make truth revelation the dominant strategy, and thus to make the allocations optimal. For example, a payment mechanism based on General Vickrey Auctions (GVA) [39] might be used to make truth revelation a weakly dominant strategy. GVA involves computation of $n+1$ winner determination problems for every combinatorial auction to calculate the agent payments. In addition to the computational complexity, the unique pricing mechanism used by the CCA protocol to ensure real-time response precludes direct adaptation of GVA mechanisms.

To understand the effect of strategic bidding, a preliminary analysis can be conducted by formulating some simple strategic bidding formulations and conducting simulation experiments. For example, Walsh et al. [41] have analyzed the effects of strategic bidding on a combinatorial auction based supply chain formation algorithm. It is important to note that the difficulty of analyzing the effects of strategic bidding actually undermines the benefits of lying about true utilities for users. An added advantage with MASM is that there is disengagement between the users and the actual sensor network.

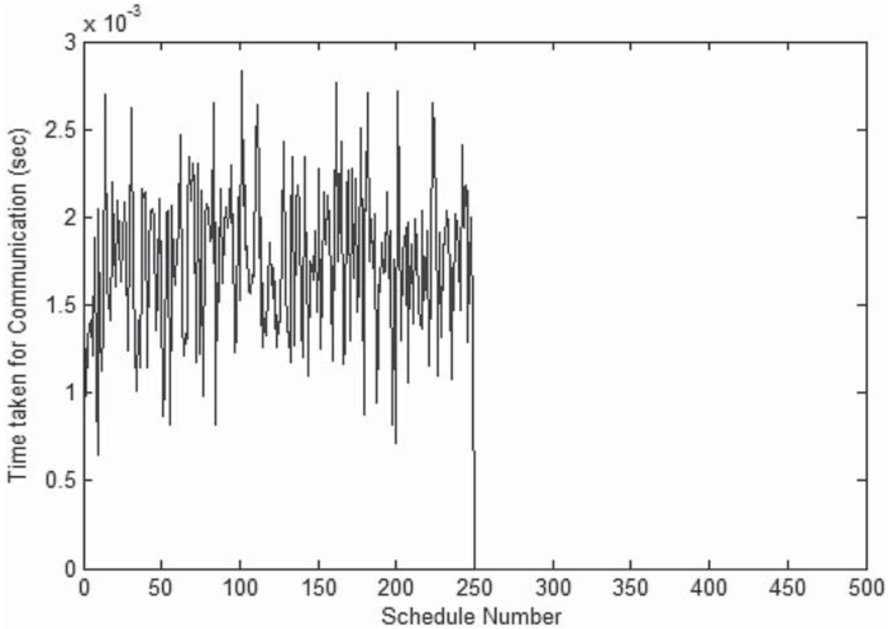


Fig. 11. Time taken for communication for a sample run vs. schedule number for a sample run with tatonnement $\dot{\tau} = 0$.

Though the users use the sensor network, the various network parameters including the prices of individual resources (which might indicate network bottlenecks) and the actual network parameters, like position of sensors, etc. is invisible to the individual user. Hence, the threat presented by malicious entities that have clandestinely gained access to use the sensor network is minimal. To analyze the effect of strategic bidding, it can be assumed that agents play Bayes-Nash strategies [19]. However, calculation of Bayes-Nash equilibria is difficult, except for the simplest of markets. An easier method for analyzing market behavior is to devise a reasonable strategic bidding policy for users and study resulting market behavior.

A simple strategic bidding policy for MASM users is to overstate their task utilities. To understand the logic behind this policy, the pricing policy of CCA should be considered. If a MASM user bids a price P for a particular task, the price it has to pay for resources allocated to its bid is not directly based on P. Instead, for any resource bundle allocated to the task during resource allocation, the user usually pays only the sum of the prices of the resources comprising the resource bundle (see round termination step in CCA). Therefore, a user that overstates its utility has the advantage of getting preferential treatment during resource allocation, while not having to pay any additional value for resources as compared to honest users. To analyze the impact of strategic bidding, experiments were conducted where a certain number of agents overstated their utility by a factor, k. The number of targets successfully destroyed by the users during the simulation experiment reflects the global performance of the market-based resource allocation. An individual user's performance is measured by its surplus defined as the difference between the total utility it obtained from destroying the targets and the total price it paid to SM for buying resources during the simulation.

TABLE IV
Market Performance with Strategic Agent Behavior

Number of strategic agents	0	2
Honest consumer surplus	1.22	−.012
Strategic consumer surplus	NA	2.3

Two sets of experiments were conducted. In the first set of experiments, all the users bid honestly. In the second set of experiments, two out of the five users overstated their utilities by a factor of two. The average surplus achieved by the honest and strategic agents is shown in Table IV. As expected, strategic agents benefited from overstating their bid prices and their average surplus increased from 1.22 to 2.3. The average surplus of the honest agents has decreased from 1.22 to -0.012 as a result of strategic bidding.

As shown by the preliminary analysis, CCA protocol encourages strategic behavior in a non-cooperative environment. However, the benefits of strategic bidding can be mitigated by using surplus sharing mechanisms where the SM charges the users a fixed percentage of their surplus on each task bid. For example, a variant of CCA, CCA-SS (Combination Auction Algorithm with Surplus Sharing) has been implemented where users are levied an additional charge of fifty percent of their expected surplus as calculated from their task bids. Under this mechanism, the overall surplus to the strategic agent decreased to -1.83. That is, they fare worse than the honest users.

Though surplus sharing provides an effective mechanism that works as a disincentive against overstating task utilities, detailed experimentation is required to analyze the complete implications of strategic behavior for some particular environment.

9. CONCLUSION

Market-based approaches provide a valuable framework for designing systems that must consider complex tradeoffs in their decision-making. Although much work has been done on individual aspects of market-based systems (e.g., auction algorithms, agent bidding strategies, etc.), there is very little work on developing a complete system in a complex real-world domain such as sensor management. Therefore, one of our contributions is to assess the design implications and components involved in building such a system. To address the issues that in the past have prevented the use of market algorithms use for sensor networks, we have developed techniques including auction mechanisms for aligning scheduling with mission objectives, approximate techniques for handling computational complexity (A-MASM), pricing mechanisms for enforcing resource constraints and surplus sharing mechanisms to reduce the impact of strategic behavior. We have also shown the system's efficiency in a simulation environment, by comparing with a weighted information-theoretic sensor manager. A crucial advantage of the proposed mechanism is its flexibility. The proposed mechanism can be easily adapted to an alternate sensor network scenario without much additional work by suitably creating QoS charts for relevant network tasks (see Fig. 2) and by adapting price equations to reflect utilization of the appropriate network resources (see Section 5). This contrasts with the current approaches based on approximate dynamic programming that are based on domain-specific and cumbersome formulation.

We are currently working on implementing these mechanisms using real-world sensor data. We are also working on developing an alternate non-myopic sensor management approach for comparison with MASM. Two possible choices are i) Maximum marginal return (MMR) [4] sensor management approach that is extended to incorporate non-myopic scheduling behavior and ii) an approximate dynamic programming based approach, similar to [37, 43, 44], that can handle multi-sensor, multi-task sensor management problems. In future, we plan to extend our market-based approach to smart dust environments, which do not have a centralized sensor manager. We also plan to extend our infrastructure to more effectively allocate information goods and develop a market-based situation assessment component. Current auction algorithms are generally designed to handle the allocation of tangible goods. However, we must adapt these e-commerce algorithms to deal with information goods in the sensor-fusion domain. Information goods (e.g., observations/reports), and the sensors/processes that generate them, may be shared between agents to effectively complete compatible tasks, where applicable. For example, if two users are engaged in the same sub-goal, and want the same information, then a single commodity can be communicated to both agents and will satisfy both of them. We plan to make use of the current research in digital auctions [10], but will need to apply it appropriately to our domain.

We would also like to develop a market-based situation assessment component that learns valuable situation assessment cues from the market bids and price information in the system. The situation assessment that users perform typically relates only to their immediate surroundings and pertains to local information only. A global perspective can be obtained by observing the overall market trends in the sensor manager's situation assessment module. For example, a sudden increase in the volume of bids from the users in one particular region of the environment could suggest an impending enemy attack in that region. Current prices can convey information about when resources are in high demand and/or scarce. As part of this effort, collaborative filtering approaches, similar to those used by Amazon [23], could mine information from a combination of goal and bid behaviors to detect strategic patterns. Eventually, one could imagine a sensor management system that recommends a new information product (e.g., a target track) based on what previous users in similar situations have selected.

REFERENCES

[1] A. Andersson and F. Ygge
Managing large scale computational markets.
In *Proceedings of the Thirty-first Hawaiian International Conference on System Sciences, Software Technology Track*, Jan. 1998, 4–13.

[2] V. Avasarala, T. Mullen, D. L. Hall and A. Garga
MASM: A market architecture for sensor management in distributed sensor networks.
In *Proceedings of the SPIE Defense and Security Symposium*, Mar. 2005, 5813–30.

[3] V. Avasarala, H. Polavarapu and T. Mullen
SGA: An approximate combinatorial auction winner determination algorithm for environments with strict real-time constraints.
In *Proceedings of the IEEE-WIC/Association for Computing Machinery (ACM) International Conference on Intelligent Agent Technology* (IAT-06), 571–578.

[4] C. F. Camerer
Behavioral Game Theory.
Princeton, NJ: Princeton University Press, 2003.

[5] D. A. Castañon
Optimal search strategies for dynamic hypothesis testing.
IEEE Transactions on Systems, Man and Cybernetics, 25 (July 1995), 1130–1138.

[6] D. A. Castañon
Approximate dynamic programming for sensor management.
In *Proceedings of the Thirty-sixth Conference on Decision and Control*, Dec. 1997, 1202–1207.

[7] S. Cheng, E. Leung, K. Lochner, K. O'Malley, D. Reeves, J. Schvartzman and M. Wellman
Walverine: A Walrasian trading agent.
Decision Support Systems, 39, 2 (Apr. 2005), 169–184.

[8] R. V. Denton, E. Alcaraz, J. Knopow, J. Llinas and K. Hintz
Towards modern sensor management systems.
In *Proceedings of the Sixth Joint Service Data Fusion Symposium*, June 14–18, 1993, 659–678.

[9] Y. Fujishima, K. Leyton-Brown and Y. Shoham
Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches.
In *Proceedings of the 16th International Joint Conference on Artificial Intelligence* (IJCAI), 548–553.

[10] A. V. Goldberg and J. D. Hartline
Envy-free auctions for digital goods.
In *Proceedings of the Fourth Association for Computing (ACM) Conference on Electronic Commerce* (EC-03), 29–35.

[11] D. L. Hall and J. Llinas
Handbook of Multisensor Data Fusion.
Boca Raton, FL: CRC Press LLC, 2001.

[12] D. L. Hall and S. McMullen
Mathematical Techniques in Multisensor Data Fusion (2nd ed.).
Norwood, MA: Artech House, 2004.

[13] K. J. Hintz and E. S. McVey
Multi-process constrained estimation.
IEEE Transaction on Systems, Man, and Cybernetics, 21, 1 (Jan./Feb. 1991), 434–442.

[14] K. J. Hintz and G. McIntyre
Goal lattices for sensor management.
In *Proceedings of the Signal Processing, Sensor Fusion and Target Recognition VIII: SPIE International Symposium on Aerospace/Defense Sensing and Control*, Apr. 1999, 249–255.

[15] K. J. Hintz and J. Malachowski
Dynamic goal instantiation in goal lattices for sensor management.
In *Proceedings of the SPIE*, May 2005, 93–99.

[16] J. S. Jordan
The competitive allocation process is informationally efficient uniquely.
Journal of Economic Theory, 28, 1 (1982), 1–18.

[17] M. Kalandros and L. Y. Pao
Covariance control for multisensor systems.
IEEE Transactions on Aerospace Electronic Systems, 38, 4 (Oct. 2002), 1138–1157.

[18] K. Kastella
Discrimination gain for sensor management in multitarget detection and tracking.
In *Proceedings of the IEEE—Systems, Man and Cybernetics (SMC)/International Association for Mathematics and Computers in Simulation (IMACS) Multi-Conference on Computational Engineering in Systems Applications* (CESA'96), July 1996, 167–172.

[19] J. D. Kattar
A solution of the multi-weapon, multi-target assignment problem.
WP-26597, The MITRE Corporation, Bedford, MA, 1986.

[20] M. Kolba and L. L. Collins
Information based sensor management in the presence of uncertainty.
IEEE Transactions on Signal Processing, 55, 6 (June 2007), 2731–2735.

[21] L.Walras
Elements of Pure Economics.
Homewood, IL: Irwin, 1954.

[22] V. Lesser, C. L. Ortiz and M. Tambe
Distributed Sensor Networks: A Multiagent Perspective.
Boston, MA: Kluwer Academic Publishers, 2003.

[23] G. Linden, B. Smith and J. York
Amazon.com recommendations: Item-to-item collaborative filtering.
IEEE Internet Computing, 7, 1 (Jan. 2003), 76–80.

[24] G. Mainland, D. C. Parkes and M. Welsh
Decentralized, adaptive resource allocation for sensor networks.
In *Proceedings of the Second The Advanced Computing Systems Association (USENIX)/Association for Computing Machinery (ACM) Symposium on Networked Systems Design & Implementation* (NSDI'05), Apr. 2005, pp. 7–13.

[25] G. A. McIntyre and K. J. Hintz
An information-theoretic approach to sensor scheduling.
In *Proceedings of the SPIE: Signal Processing, Sensor Fusion, and Target Recognition V*, Apr. 1996, 304–312.

[26] G. A. McIntyre and K. J. Hintz
Sensor management simulation and comparative study.
In *Proceedings of the SPIE International Symposium on Aerospace/Defense Sensing & Control: Signal Processing, Sensor Fusion, and Target Recognition VI*, Apr. 1997.

[27] J. Moffat
Complexity Theory and Network Centric Warfare.
CCRP Publication Series, 2003.

[28] T. Mullen, V. Avasarala and D. L. Hall
Bringing the market to level 4 fusion: Investigation of auction methods for sensor resource allocation.
In *Proceedings of the Military Sensing Symposia (MSS) National Symposium on Sensors and Data Fusion* (NSSDF), May 2005.

[29] T. Mullen, V. Avasarala and D. L. Hall
Customer-driven sensor management.
IEEE Intelligent Systems, 21, 2 (2006), 41–49.

[30] R. Muraleedharan and L. Osadciw
Decision in a building access system using swarm intelligence and posets.
In *Proceedings of the Conference on Information Sciences and Systems*, Mar. 2004, 414–419.

[31] J. M. Nash
Optimal allocation of tracking resources.
In *Proceedings of the IEEE Conference on Decision and Control*, Dec. 1977, 1177–1180.

[32] J. Ostwald, V. Lesser and S. Abdallah
Combinatorial auction for resource allocation in a distributed sensor network.
In *Proceedings of the The 26th IEEE International Real-Time Systems Symposium* (RTSS'05), Dec. 2005, pp. 266–274.

[33] P. Padhy, R. K. Dash, K. Martinez and N. R. Jennings
A utility-based sensing and communication model for a glacial sensor network.
In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems* (AAMAS'06), May 2006, 1353–60.

[34] P. A. Samuelson
Foundations of Economic Analysis.
Cambridge, MA: Harvard University Press, 1947.

[35] T. Sandholm
Algorithm for optimal winner determination in combinatorial auctions.
Artificial Intelligence, 135, 1–2 (Feb. 2002), 1–54.

[36] W. Schmaedeke
Information-based sensor management.
In *Proceedings of the SPIE: Signal Processing, Sensor Fusion, and Target Recognition II*, Apr. 1993, 156–64.

[37] M. K. Schneider and C.-Y. Chong
A rollout algorithm to coordinate multiple sensor resources to track and discriminate targets.
In *Proceedings of the SPIE Conference on Signal Processing, Sensor Fusion and Target Recognition*, Apr. 2006.

[38] J. B. Shoven and J. Whalley
Applying General Equilibrium.
New York: Cambridge University Press, 1992.

[39] H. R. Varian and J. K. MacKieMason
Generalized Vickrey auctions.
Department of Economics, University of Michigan, 1994.

[40] C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart and S. Stornetta
Spawn: A distributed computational economy.
IEEE Transactions on Software Engineering, 18, 103–117.

[41] W. E. Walsh, M. P. Wellman and F. Ygge
Combinatorial auctions for supply chain formation.
In *Proceedings of the Association for Computing Machinery (ACM) Conference on Electronic Commerce* (EC'00), Oct. 2000, 260–269.

[42] W. E. Walsh, M. P. Wellman, P. R. Wurman and J. K. MacKie-Mason
Some economics of market-based distributed scheduling.
In *Proceedings of the International Conference on Distributed Computing Systems*, May 1998, 612–621.

[43] R. Washburn, M. Schneider and J. Fox
Stochastic dynamic programming based approaches to sensor resource management.
In *Proceedings of the Fifth International Conference on Information Fusion*, 608–615.

[44] J. L. Williams, J. W. Fisher, III and A. S. Willsky
An approximate dynamic programming approach to a communication constrained sensor management problem.
In *Proceedings of the Eighth International Conference on Information Fusion*, July 2005.

[45] F. Ygge
Market-Oriented Programming and its Application to Power Load Management.
Ph.D. thesis, Department of Computer Science, Lund University, ISBN 91-628-3055-4, CODEN LUNDFD6/(NFCS-1012)/1-224/(1998).

Viswanath Avasarala obtained his bachelor's degree from the Indian Institute of Technology, Madras in 2001. He has a Ph.D. in information sciences and technology from Pennsylvania State University. In his Ph.D. thesis, he investigated the use of market-oriented multi-agent frameworks for real-time and distributed sensor management.

His primary research interests are in data mining, market-oriented programming and multi-agent collaboration. Since 2006, he has been working as an information scientist at GE Global Research. At GE, Dr. Avasarala uses artificial intelligence based methodologies to solve research problems in machinery diagnostics and prognostics, fleet logistics and health management, stochastic simulation, multi-objective optimization and medical imaging. He has published 3 journal papers, 1 book chapter and more than 10 conference papers in these areas. He has also chaired and/or acted as an organizing committee member of several reputed international conferences in soft computing. He is an active IEEE member and the current chair of Schenectady chapter of the IEEE Computational Intelligence Society.

Tracy Mullen is an assistant professor at the College of Information Sciences and Technology at The Pennsylvania State University. She received her Ph.D. in computer science from the University of Michigan.

She has previously worked at Lockheed Martin, Bellcore, and NEC Research. Dr. Mullen's research is focused on computational markets and multiagent systems to support decentralized decision making. Specific interests include market-based resource allocation for sensor management, prediction markets, and trading agents and markets. Her research papers have been published in *ACM Transactions on Internet Technology*, *Decision Support Systems*, *Electronic Commerce Research*, *IEEE Computer*, *Mathematics and Computers in Simulation*, and *Operating Systems Review*, among others.

David L. Hall is a professor in the Penn State College of Information Sciences and Technology (IST) where he leads the Center for Network Centric Cognition and Information Fusion. Previously he was the associate dean for Research and Graduate Programs in Penn State's College of Information Sciences and Technology, where he was responsible for overseeing IST's Ph.D. and Master's Degree programs as well as research grant administration and leadership of IST faculty research efforts. Hall joined IST from the University's Applied Research Laboratory (ARL) where he served as associate director and senior scientist. At ARL, he oversaw the 150-person Information and Network Systems Office, comprised of four divisions: Information Science and Technology, Navigation Research and Development, Systems and Operations Automation, and Communications Science and Technology. He has more than 25 years of experience in research, research management, and systems development.

He is the author of more than 200 papers, reports, books and book chapters, and he has delivered numerous lectures on his research, research management, and artificial intelligence. Hall has been named as an IEEE fellow for his contributions to data fusion and he was awarded the DoD Joe Mignona National Data Fusion award for his contributions and leadership in data fusion.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

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

2009 BOARD OF DIRECTORS*

2007–2009	**2008–2010**	**2009–2011**
Stefano Coraluppi	Steve Andler	Éloi Bossé
Jean Dezert	Subhash Challa	Uwe Hanebeck
Wolfgang Koch	Mahendra Mallick	Roy Streit

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

PAST PRESIDENTS

Darko Musicki, 2008	Chee Chong, 2004	Yaakov Bar-Shalom, 2000
Erik Blasch, 2007	Xiao-Rong Li, 2003	Jim Llinas, 1999
Pierre Valin, 2006	Yaakov Bar-Shalom, 2002	Jim Llinas, 1998
W. Dale Blair, 2005	Pramod Varshney, 2001	

SOCIETY VISION

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

SOCIETY MISSION

Advocate

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

Serve

To serve its members and engineering, business, and scientific communities by providing high-quality information, educational products, and services.

Communicate

To create international communication forums and hold international conferences in countries that provide for interaction of members of fusion communities with each other, with those in other disciplines, and with those in industry and academia.

Educate

To promote undergraduate and graduate education related to information fusion technologies at universities around the world. Sponsor educational courses and tutorials at conferences.

Integrate

Integrate ideas from various approaches for information fusion, and look for common threads and themes–look for overall principles, rather than a multitude of point solutions. Serve as the central focus for coordinating the activities of world-wide information fusion related societies or organizations. Serve as a professional liaison to industry, academia, and government.

Disseminate

To propagate the ideas for integrated approaches to information fusion so that others can build on them in both industry and academia.

Call for Papers

The Journal of Advances in Information Fusion (JAIF) seeks original contributions in the technical areas of research related to information fusion. Authors of papers in one of the technical areas listed on the inside cover of JAIF are encouraged to submit their papers for peer review at http://jaif.msubmit.net.

Call for Reviewers

The success of JAIF and its value to the research community is strongly dependent on the quality of its peer review process. Researchers in the technical areas related to information fusion are encouraged to register as a reviewer for JAIF at http://jaif.msubmit.net. Potential reviewers should notify via email the appropriate editors of their offer to serve as a reviewer.

Journal of Advances in Information Fusion Volume 4 Number 1 June 2009 ISSN 1557-6418