SWEDISH DEFENCE
RESEARCH AGENCY

FOI-R--0158--SE
July 2001
ISSN 1650-1942
Scientific report

Ning Xiong and Per Svensson

Sensor Management for Information Fusion - A Review

Command and Control Warfare Technology
SE-581 11 Linköping

SWEDISH DEFENCE RESEARCH AGENCY
Command and Control Warfare Technology
P.O. Box 1165
SE-581 11 Linköping

FOI-R--0158--SE
July 2001
ISSN 1650-1942
Scientific report

Ning Xiong and Per Svensson

Sensor Management for Information Fusion - A Review

Issuing organization	Report number, ISRN	Report type
FOI – Swedish Defence Research Agency Command and Control Warfare Technology P.O. Box 1165 SE-581 11 Linköping	FOI-R--0158--SE	Scientific report
	Research area code 4. C4ISR	
	Month year July 2001	**Project no.** E 7037
	Customers code 5. Contracted Research	
	Sub area code 41 C4I	
Author/s (editor/s) Ning Xiong and Per Svensson	**Project manager** Per Svensson	
	Approved by	
	Scientifically and technically responsible Per Svensson	

Report title

Sensor Management for Information Fusion - A Review

Abstract (not more than 200 words)

Multi-sensor management concerns the control of environment perception activities by managing or coordinating the usage of multiple sensor resources. It is an emerging research area, which has become increasingly important in research and development of modern multi-sensor systems. This article presents a comprehensive review of multi-sensor management in relation to multi-sensor information fusion, describing its place and role in the larger context, generalizing main problems from existing application needs, and highlighting problem solving methodologies.

Keywords

sensor management, multi-sensor system, data fusion, information fusion, sensor deployment, sensor behavior assignment, sensor coordination.

Further bibliographic information	Language English
ISSN 1650-1942	**Pages** 47 p.
	Price acc. to pricelist **Security classification: None**

Utgivare Totalförsvarets Forskningsinstitut - FOI Ledningssystemteknik Box 1165 581 11 Linköping	**Rapportnummer, ISRN** FOI-R--0158--SE	**Klassificering** Vetenskaplig rapport
	Forskningsområde 4. Spaning och ledning	
	Månad, år Juli 2001	**Projektnummer** E 7037
	Verksamhetsgren 5. Uppdragsfinansierad verksamhet	
	Delområde 41 Ledning med samband och telekom och IT-	
Författare/redaktör Ning Xiong och Per Svensson	**Projektledare** Per Svensson	
	Godkänd av	
	Tekniskt och/eller vetenskapligt ansvarig Per Svensson	

Rapportens titel (i översättning)

Sensorstyrning för Informationsfusion - en översikt

Sammanfattning (högst 200 ord)

Begreppet sensorstyrning avser metodik för informationsinsamling som bygger på styrning och samordning av sensorresurser. Det är ett område av växande betydelse inom forskning och utveckling kring moderna multisensorsystem.

Denna rapport presenterar en översikt av hur sensorstyrning kan tillämpas inom informationsfusion, beskriver sensorstyrningens plats och roll i detta sammanhang samt abstraherar centrala metodfrågor utifrån kända tillämpningsbehov. Rapporten ger också en översikt över problemlösningsmetoder inom området.

Nyckelord

Se

Övriga bibliografiska uppgifter	**Språk** Engelska
ISSN 1650-1942	**Antal sidor:** 47 s.
Distribution enligt missiv	**Pris: Enligt prislista** **Sekretess: Öppen**

Contents

1. Introduction	**5**
1.1 The Fundamental Purpose of Sensor Management	5
1.2 The Role of Sensor Management in Information Fusion	6
1.3 Multi-Sensor Management Architectures	7
1.4 Classification of Multi-Sensor Management Problems	8
1.4.1 Sensor deployment	9
1.4.2 Sensor behavior assignment	9
1.4.3 Sensor coordination in a decentralized sensor network	10
1.5 Organization of the Paper	11
2. A Perspective on Sensor Management Problem Solving	**12**
2.1 Principles and Methodological Considerations	12
2.2 Top-Down Problem Solving by a Sensor Manager	15
3. Principles of Sensor Placement	**18**
3.1 General Description	18
3.2 Signal Filtering Related to Sensor Placement	19
4. Observation Task Evaluation	**21**
4.1 Evaluation Based on Fuzzy Decision Trees	21
4.2 Evaluation Using Neural Networks	22
4.3 Evaluation Based on Goal Lattices	23
5. Measurement Policies for Information Collection	**26**
5.1 Which Measurement Pattern to Use	26
5.2 When to Observe a Tracked Target	26
5.3 In What Sequence to Detect an Unknown Target	28
6. Sensor Resource Allocation	**30**
6.1 Search-Based Approaches to Sensor Selection	30
6.2 Information-Theoretic Approaches to Sensor Selection	33
6.3 Decision-Theoretic Approaches to Sensor Planning	35
6.4 Fuzzy Logic Resource Manager	36
6.5 Markov Decision for Sensor Allocation in Classification	36
7. Towards Cooperative Sensor Behavior	**40**
8. Conclusion and Discussion	**42**
References	**44**

1 Introduction

Multi-sensor systems are becoming increasingly important in a variety of military and civilian applications. Since a single sensor generally can only perceive limited partial information about the environment, multiple similar and/or dissimilar sensors are required to provide sufficient local pictures with different focus and from different viewpoints in an integrated manner. Further, information from heterogeneous sensors can be combined using data fusion algorithms to obtain synergistic observation effects. Thus, the benefits of multi-sensor systems are to broaden machine perception and enhance awareness of the state of the world compared to what could be acquired by a single sensor system.

With the advancements in sensor technology, sensors are becoming more agile. Also, more of them are needed in a scenario in response to the increasingly intricate nature of the environment to be sensed. The increased sophistication of sensor assets along with the large amounts of data to be processed has pushed the information acquisition problem far beyond what can be handled by a human operator. This motivates the emerging interest in research into automatic and semi-automatic management of sensor resources for improving overall perception performance beyond basic fusion of data.

1.1 The Fundamental Purpose of Sensor Management

Multi-sensor management is formally described as a system or process that seeks to manage or coordinate the usage of a suite of sensors or measurement devices in a dynamic, uncertain environment, to improve the performance of data fusion and ultimately that of perception. It is also beneficial to avoid overwhelming storage and computational requirements in a sensor and data rich environment by controlling the data gathering process such that only the truly necessary data are collected and stored [60]. The why and what issues of both single-sensor and multi-sensor management were thoroughly discussed in the papers ([1], [7], [53-54], [56]). To reiterate, the basic objective of sensor management is to select the right sensors to do the right service on the right object at the right time. The sensor manager is responsible for answering questions like:

- Which observation tasks are to be performed and what are their priorities?
- How many sensors are required to meet an information request?
- When are extra sensors to be deployed and in which locations?
- Which sensor sets are to be applied to which tasks?
- What is the action or mode sequence for a particular sensor?
- What parameter values should be selected for the operation of sensors?

The simplest job of sensor management is to choose the optimal sensor parameter values given one or more sensors with respect to a given task, see for example the paper [72]. This is also called active perception where sensors are to be configured optimally for a specific purpose. More general problems of (multi-)sensor management are, however, related to decisions about what sensors to use and for which purposes, as well as when and where to use them. Widely acknowledged is the fact that it is not realistic to continually observe everything in the environment and therefore selective perception becomes necessary, requiring the sensor management system to decide when to sense what and with which sensors. Typical temporal complexities which must be accommodated in the sensor management process were discussed in [46].

1.2 The Role of Sensor Management in Information Fusion

Sensor management merits incorporation in information fusion processes. Although terminology has not yet fully stabilized, it is generally acknowledged that information fusion is a collective concept comprising situation assessment (level 2), threat or impact assessment (level 3) and process refinement (level 4) in the so-called JDL model of data fusion [76]. As pointed out in [70], in addition to intelligence interpretation, information fusion should be equipped with techniques for proactive or reactive planning and management of own collection resources such as sensors and sensor platforms, in order to make best use of these assets with respect to identified intelligence requirements. Sensor management, aiming at improving data fusion performance by controlling sensor behavior, plays the role of level 4 functions in the JDL model.

Sensor management indeed provides information feedback from data fusion results to sensor operations ([7], [54]). The representation of the data fusion process as a feedback closed-loop structure is depicted in Fig. 1, where the sensor manager on level 4 uses the information from levels 0-3 to plan future sensor actions. The feedback is intended to improve the data collection process with expected benefits of earlier detection, improved tracking, and more reliable identification, or to confirm what might be tactically inferred from previously gathered evidences. Timeliness is a necessary requirement on the feedback management of sensors for fast adaptation to environment changes. That is to say, a prompt decision on sensor functions has to be made before the development of the tactical situation has made such a decision obsolete.

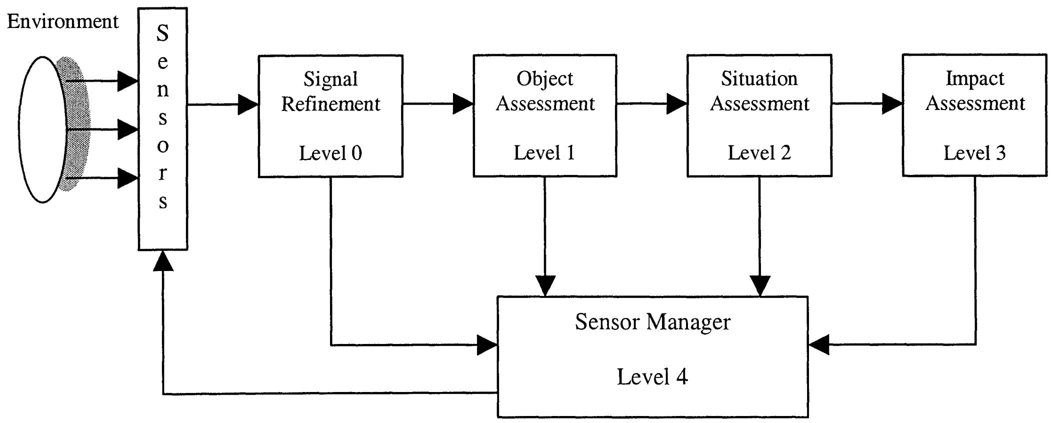

Fig. 1. Feedback connection via sensor manager in a data fusion process

As a categorization of process refinement, Steinberg et al [68] classified responses of resources (including sensors) as *reflexive*, *feature-based*, *entity-relation based*, *context-sensitive*, *cost-sensitive*, and *reaction-sensitive*, in terms of input data/information types. This categorization is viewed as an expansion of Dasarathy's model [18] in which data fusion functions are subdivided considering merely data, features and objects as possible input/output types.

1.3 Multi-Sensor Management Architectures

The architecture of a multi-sensor management system is closely related to the form of data fusion unit. Typically there are three alternatives for system structure, namely:

1) ***Centralized***. In a centralized system the data fusion unit is treated as a central mechanism. It collects information from all different platforms and sensors and decides jobs that must be accomplished by individual sensors. All commands sent from the fusion center to respective sensors must be accepted and followed with proper sensor actions.

2) ***Decentralized***. In a decentralized system data are fused locally with a set of local agents rather than by a central unit. In this case, every sensor or platform can be viewed as an intelligent asset having some degree of autonomy in decision-making. Sensor coordination is achieved based on communication in the network of agents, in which sensors share locally fused information and cooperate with each other. Durrant-Whyte et al [23] stated that decentralized data fusion exhibits many attractive properties by being:

- scalable in structure without being constrained by centralized computational bottlenecks or communicational bandwidth limitations;
- survivable in the face of on-line loss of sensing nodes and to dynamic changes of the network;
- modular in the design and implementation of fusion nodes.

However, the effect of redundant information is a serious problem that may arise in decentralized data fusion networks [16]. It is not possible, within most filtering frameworks, to combine information pieces from multiple sources unless they are independent or have known cross-covariance [36]. Moreover, without any common communication facility, data exchange in such a network must be carried out strictly on a node-to-node basis. A delay between sender and receiver could result in transient inconsistencies of the global state among different parts of the network, causing degradation of overall performance [28].

3) ***Hierarchical***. This can be regarded as a mixture of centralized and decentralized architectures. In a hierarchical system there are usually several levels of hierarchy in which the top level functions as the global fusion center and the lowest level consists of several local fusion centers [56]. Every local fusion node is responsible for management of a sensor subset. The partitioning of the whole sensor assembly into different groups can be realized based on either sensors' geographical locations or platforms, sensor functions performed, or sensor data delivered (to ensure commensurate data from the same sensor group).

Two interesting instances of sensor management architectures are given in the following for illustration.

The macro/micro architecture proposed by [7] can be classified as a two-level hierarchical system. It consists of a macro sensor manager playing a central role and a set of micro sensor managers residing with respective sensors. The macro sensor manager is in charge of high-level strategic decisions about how to best utilize the available sensing resources to achieve the mission objectives. The micro sensor manager schedules the tactics of a particular sensor to best carry out the requests from the macro manager. Thus it is clear that every managed sensor needs its own micro manager.

Another hybrid distributed and hierarchical approach was suggested in [60] for sensor-rich environments exemplified by an aircraft health and usage monitoring system. The main idea is to distribute the management function across system functional or physical boundaries with global oversight of mission goals and information requests. One such model for management of numerous sensors is shown in Fig. 2. At the top of the model is the *mission manager* tasked with converting mission goals to information needs, which are then mapped by the *information instantiator* into a set of measurement patterns in accordance with those needs. The role of the *meta-manager* is to enable natural subdivision of a single manager into a set of mostly independent *local resource managers* each being responsible for a particular sensor subset. Occasionally, these local managers need to be coordinated by the meta-manager if there is a request for information, which cannot be satisfied by a single sensor suite. A major difficulty in implementing this hybrid architecture of sensor management lies with the meta-manager. It is not yet obvious how to best translate global functional needs into a set of local resource managers and how to coordinate the disparate local managers distributed across functional or physical boundaries.

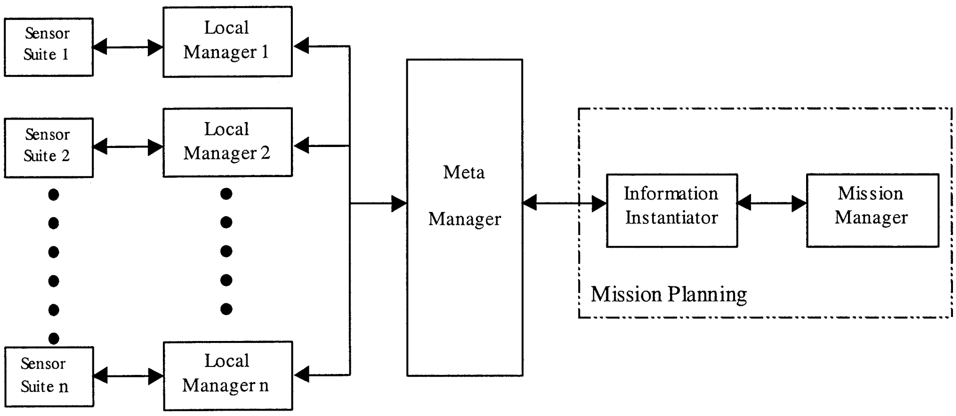

Fig. 2. A distributed and hierarchical sensor management model for a sensor rich environment (cited from [60])

1.4 Classification of Multi-Sensor Management Problems

Multi-sensor management is a broad concept referring to a set of distinct issues of planning and control of sensor resource usage to enhance multi-sensor data fusion performance. Various aspects of this area have been discussed in papers in the open literature. Generally, these problems fall into three main categories, i.e., sensor deployment, sensor behavior assignment, and sensor coordination.

1.4.1 Sensor Deployment

Sensor deployment is a critical issue for intelligence collection in an uncertain dynamic environment. It concerns making decisions about when, where, and how many sensing resources need to be deployed in reaction to the state of the world and its changes. In some situations, it should be beneficial to proactively deploy sensing resources according to a predicted situation development tendency in order to get prepared to observe an event, which is likely to happen in the upcoming period.

Sensor placement [59] needs special attention in sensor deployment. It consists of positioning multiple sensors simultaneously in optimal or near optimal locations to support surveillance tasks when necessary. Typically it is desired to locate sensors within a particular region determined by tactical situations to optimize a certain criterion usually expressed in terms of global detection probability, quality of tracks, etc. This problem can be formulated as one of constrained optimization of a set of parameters. It is subject to constraints due to the following factors:

- Sensors are usually restricted to specified regions due to tactical considerations.
- Critical restrictions may be imposed on relative positions of adjacent sensors to enable their mutual communication when sensors are arranged as distributed assets in a decentralized network.
- The amount of sensing resources that can be positioned in a given period is limited due to logistical restrictions.

In simple cases, decisions on sensor placement are to be made with respect to a well-prescribed and stationary environment. As examples, we may consider such application scenarios as:

- Placing radars to minimize the terrain screening effect in detection of an aircraft approaching a fixed site.
- Arrangement of a network of intelligence gathering assets in a specified region to target another well-defined area.

In the above scenarios mathematical or physical models such as terrain models, propagation models, etc. are commonly available and they are used as the basis for evaluation of sensor placement decisions.

More challenging are those situations in which the environment is dynamic and sensors must repeatedly be repositioned to be able to refine and update the state estimation of moving targets in real time. Typical situations where reactive sensor placement is required are:

- Submarine tracking by means of passive sonobuoys in an anti-submarine warfare scenario.
- Locating moving transmitters using ESM (electronic support measures) receivers.
- Tracking of tanks on land by dropping passive acoustic sensors.

1.4.2 Sensor Behavior Assignment

The basic purpose of sensor management is to adapt sensor behavior to dynamic environments. By *sensor behavior assignment* is meant efficient determination and planning

of sensor functions and usage according to changing situation awareness or mission requirements. Two crucial points are involved here:
1) Decisions about the set of observation tasks (referred as system-level tasks) that the sensor system is supposed to accomplish currently or in the near future, on grounds of the current/predicted situation as well as the given mission goal.

2) Planning and scheduling of actions of the deployed sensors to best accomplish the proposed observation tasks and their objectives.

Owing to limited sensing resources, it is prevalent in real applications that available sensors are not able to serve all desired tasks and achieve all their associated objectives simultaneously. Therefore a reasonable compromise between conflicting demands is sought. Intuitively, more urgent or important tasks should be given higher priority in their competition for resources. Thus a scheme is required to prioritize observation tasks. Information about task priority can be very useful in scheduling of sensor actions and for negotiation between sensors in a decentralized paradigm.

To concretize this class of problems, let us consider a scenario including a number of targets as well as multiple sensors, which are capable of focusing on different objects with different modes for target tracking and/or classification. The first step for the sensor management system should be to utilize evidences gathered to decide objects of interest and to prioritize which objects to look at in the time following. Subsequently, in the second step, different sensors together with their modes are allocated across the interesting objects to achieve best situation awareness. In fact, owing to the constraints on sensor and computational resources, it is in general not possible to measure all targets of interest with all sensors in a single time interval. Also, improvement of the accuracy on one object may lead to degradation of performance on another object. What is required is a suitable compromise among different targets.

It is worth noting that although several distinct terms appear in the literature such as *sensor action planning* in [41], *sensor selection* in ([25], [37-38]), as well as *sensor-to-task assignment* in [52-53], these terms inherently signify the same aspect of distributing resources among observation tasks, thus belong to the second issue of this problem class. In this paper we present the more general concept *sensor behavior assignment,* which involves not only the arrangement of operations for individual sensors but also inferences about system-level tasks and objectives to be accomplished. Actually, specification of tasks at the system level can be considered as postulating expected overall behaviors of the perception system as a whole, while planning and scheduling of sensor actions defines local behaviors residing with specific sensors. Dynamic information associated with time-varying utility and availability serves here as the basis for decision making about sensor behaviors.

1.4.3 Sensor Coordination in a Decentralized Sensor Network

There are two general ways to integrate a set of sensors into a sensor network. One is the centralized paradigm, where all actions of all sensors are decided by a central mechanism. The other alternative is to treat sensors in the network as distributed intelligent agents with some degree of autonomy [74]. In such a decentralized architecture, bi-directional communication between sensors is enabled, so that communication bottlenecks possibly existing in a centralized network can be avoided. A major research objective of decentralized sensor management is to establish cooperative behavior between sensors with no or little external supervision.

An interesting scenario requiring sensor coordination is shown in Fig. 3 where five autonomous sensors cooperatively explore an area of interest. The sensing ranges of the sensors are shown as shaded circles in the figure and *Vi* denotes the velocity vector of sensor *i*. That is, every sensor has its own dynamics, can perceive only part of the area and thus has its own local view of the world. These local views can be shared by some members of the sensor community. Intuitively, a local picture from one sensor can be used to direct the attention of other sensors or transfer tasks such as target tracking from one sensor to another. An interesting question is how participating sensors can autonomously coordinate their movements and sensing actions, on grounds of shared information, to develop an optimal global awareness of the environment with parsimonious consumption of time and resources.

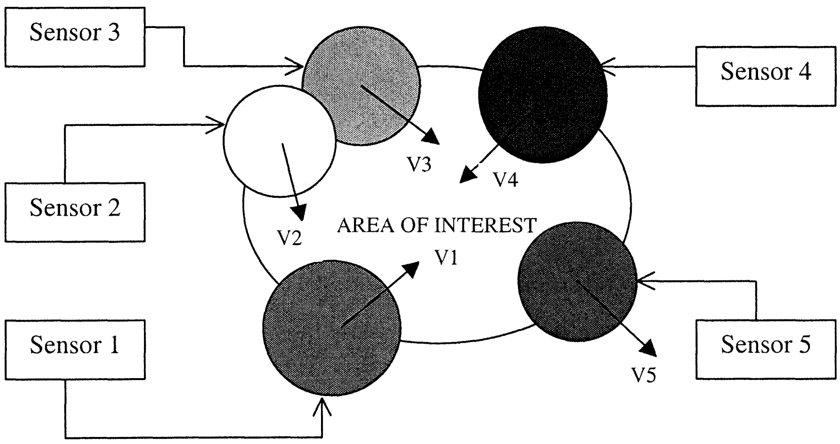

Fig. 3. A team of mobile sensors cooperatively observing an area of interest

1.5 Organization of the Paper

Following the overview of multi-sensor management issues given above, Section 2 of this paper presents a general perspective on sensor management problem solving. We will discuss principles and methodological foundations upon which sensor management systems may be based and then suggest a hierarchical top-down procedure for sensor managers to solve a complex perception control problem requiring comprehensive functionality.

Nevertheless, there is no single mechanism capable of accomplishing all functional activities of all levels of the sensor management hierarchy. Although some published papers in the open literature incorporate the term sensor management in their titles, only one topic within this general field is usually discussed by each individual paper. In the following sections, we will therefore discuss known approaches for sensor management in terms of separate issues such as principles of sensor placement (Section 3), observation task evaluation (Section 4), measurement policies for information collection (Section 5), sensor resource allocation (Section 6), and sensor behavior cooperation (Section 7). Section 8 concludes the paper and discusses issues for future research.

We will not dwell on sensor parameter control and sensor job (time) scheduling, since these topics seem more relevant to single-sensor rather than multi-sensor systems on which our paper is mainly focused. Approaches to sensor scheduling will be briefly mentioned when discussing level 1 management activities in the top-down problem solving procedure (Section 2).

2 A Perspective on Sensor Management Problem Solving

2.1 Principles and Methodological Considerations

Inherently the purpose of sensor management is to optimize the data fusion performance by feedback control of sensor resources. The performance index was called *figure of merit* in [7]. Its definition depends on what one wishes to optimize, typical quantities including probability of target detection, track/identification accuracy, probability of loss-of-track, probability of survival, probability of target kill, etc. Basically, we desire the performance index established for optimization to transcend the diversity of sensors and to be analytically/computationally tractable. Further, in order to avoid potentially myopic sensor management strategies, a prospective figure of merit should take into account both short-term and long-term interests so as to arrive at a good balance in final outcomes. Optimization according to long-term objectives is critical to the development of global improvements of the perception process with respect to an evolving scenario. Recent interesting research efforts to realize non-myopic sensor management functionality can be found in ([35], [71], [73], [77]).

From the viewpoint of decision theory, sensor management is a decision making task to determine the most appropriate sensor action to perform in order to achieve maximum utility. Such a decision problem was treated in [41] based upon a Bayesian decision tree where distinct perception actions were assessed against each other in terms of combined costs and benefits. Fung et al [24] and Musick et al [54] discussed the applicability of influence diagrams as a graphical modeling technique in support of decision analysis for the selection of sensor action. Decision making for sensor management tends to be a sequential process in the sense that intermediate decisions have to be made while a dynamic situation evolves and where penalties and awards of decisions made are only revealed over time. It has been recognized that in many cases sensor management can be modeled as a Markov decision problem subject to uncertainty, solutions to which were studied by Castanon [14-15] based on stochastic dynamic programming. Although dynamic programming offers a mathematically tractable structure to find the optimum sequence of decisions, Castanon's approach is likely to suffer from combinatorial explosion when solving practical problems of even moderate size.

By viewing sensors as constrained communication channels [33], sensor management is intended to control the functions of such channels to provide the maximum quantity of useful information within a limited time period. That is to say, we would like to optimize the information passed through sensor channels by proper arrangement of their operations. Since uncertainty about the environment is reduced by means of a measurement, each sensor action has the potential merit of contributing information as it is performed. The key point here is how to assess the relative merits of candidate sensing plans in terms of information gained or uncertainty reduced. One direct means serving this purpose is to borrow the concept of entropy from information theory as a measure of the uncertainty about the state of the environment. In this way, we can quantify the amount of information gained due to accomplishment of a sensing plan as either entropy change (in Shannon's definition) or cross-entropy (in Kullback-Leibler's definition), see Section 6.2.

In recent years, entropy-based information metrics have been adopted in many studies of sensor management in various scenarios, including: controlling a single sensor to track multiple targets [33], sensor-target pairings in multi-sensor and multi-target tracking ([19], [61-62]), search area determination [39], and search versus track trade-offs in a simulated environment [49]. However, all these papers consider only the amount of information rather than the value of information. They aim to maximize the quantity of information provided by sensors without taking into account whether the information derived is useful or interesting. Occasionally, this might lead to directing sensor attention to be paid to non-significant aspects of the environment in order to reach a superficial maximum of information attainment. Further research is thus needed to develop more comprehensive information measures, in order to drive sensor management towards optimization in terms of the utility of obtained information with respect to mission goals rather than merely the gained amount of information [3]. Regarding recent efforts in this direction we note the attempts by ([32], [48]), in which goal lattices were employed to prioritize observation tasks in terms of their mission-accomplishing value rather than the quantity of obtained information.

As sensor management plays the role of feedback controlling sensor resources, the development of optimal control laws and strategies suitable for real applications is a challenging problem. Pre-designing a management algorithm ensuring best control behavior at all times is desirable but can be extremely difficult due to high uncertainty and complexity of the environment to be sensed. A perhaps more feasible way of achieving optimality of sensor management performance is to incorporate machine learning techniques to enable sensor managers to learn from experience. In principle, there are two general classes of machine learning useful to arrive at optimal management strategies, namely off-line learning and on-line learning. Off-line learning, as practiced in ([40], [65-66]), attempts to generalize valuable control (management) knowledge from a sufficient amount of examples. It is actually a form of inductive learning, which desires a comprehensive training database of scenarios covering a wide range of situations. Using a database with only a small number of scenarios could result in a sensor manager that will be effective in some cases but ineffective in others. On the other hand, on-line learning represents the idea of the sensor manager directly learning to manage sensing assets through its interactions with the environment. This mode of learning is more difficult to realize than off-line learning in that an assessment signal has to be created for every sensing action conducted, in order for management strategies to be immediately updated upon completion of a perception. In particular, *reinforcement learning* techniques ([8], [47]) were found effective for on-line learning of sensor search strategies for static targets. Generally speaking, despite some preliminary work done (as mentioned above), machine learning for sensor management is still an under-researched area and much room remains for further study in this direction.

In addition, in the authors' view, behavior-based artificial intelligence [13] could be a useful asset in information fusion process refinement. As it is known that behavior-based systems [2] exhibit strong robustness and flexibility against a dynamically changing world, incorporating reactive skills in sensor management would enable the perception system to have reflexes for coping with eventual deviations and to recover instantaneously from unexpected events. In realizing this purpose, a deliberative-reactive architecture needs to be established in the process refinement level for integration of abstract planning and behavior-based reactive control in a coherent manner.

2.2 Top-Down Problem Solving by a Sensor Manager

As we know, sensor management is a complex procedure intended to guide the information collection process with respect to a broad range of activities. It is not realistic for a sensor manager to solve its complete problem by a single mechanism, because the huge and perplexing space of possible alternatives tends to make this goal infeasible in real time. On the other hand, problem solving for sensor management can be made much easier and more effective by partitioning tasks into activities along different layers. Thus, a sensor manager might follow a top-down policy and proceed step by step from meta-issues to detailed reasoning. The following five activity levels would be characteristic of a sensor management system having a comprehensive functionality along these lines:

Level 4 (*mission planning*). This is the highest level of sensor management responsible for deciding system-level tasks based on information generated internally or externally. It concerns meta-sensor management issues such as:
- Which services to perform (e.g. search, target tracking or target updating)?
- Which accuracy level (e.g. desirable error covariance) to aim at?
- How frequently to measure?
- On which area of the environment to focus?
- Which targets to select?
- How to rank the importance or priorities of the required tasks?

Mission planning plays the role of indirectly managing the behavior of the sensor system but does not deal with implementation details. Meta-reasoning is required here, to be conducted according to results from other data fusion functions. At the same time it must also provide access to human operators, enabling them to impose their special requests when necessary.

Level 3 (*resource deployment*). This management level is needed for surveillance purposes in a dynamic, uncertain environment. In such cases, new extra sensing devices have to be deployed whenever necessary in order to upgrade sensing capabilities or to catch up with quick environment changes. This involves proactive or reactive planning of sensor assets to be deployed in a scenario. More concretely, the level of resource deployment is responsible for answering questions such as:
- When are extra sensors required and how many?
- Where to place the newly required sensors?

Level 2 (*resource planning*). The purpose of this level is to propose requests on individual sensors by deciding what jobs/actions are expected to be performed by them. Typical activities on this level are:
- Sensor selection for multi-sensor multi-target tracking;
- Sensor allocation for simultaneous classification of numerous objects;
- Sensor cueing (sensor handing over, target acquisition by another aiding sensor);
- Movement planning for mobile sensors and platforms;

• Negotiation and cooperation between sensors in a decentralized sensor network.

Level 1 (*sensor scheduling*). This level has the role to set up a timeline of commands for every sensor to carry out based on sensor availability and capabilities, given job/action requests from level 2. Each sensor is assigned with a detailed schedule on what to do in each time interval. This is a non-trivial problem considering the distinct characteristics of jobs/actions to be performed, such as hard or fluid deadlines, as well as varying job/action priorities. As important solutions to scheduling we refer to such methods as *brick packing* [7], *best first* [7], *genetic search* [17], *OGUPSA* [78], and the enhanced version of *OGUPSA* [51] with its usage demonstrated in a sensor management simulation [50].

Level 0 (*sensor control*). At the lowest location in the hierarchy, this level is in charge of controlling every degree of freedom of the sensor under a given command. It involves definition of all parameters that specify the details for realization of that command. For example, a multi-functional radar accepts general commands such as search, target tracking or target updating. Each of these command categories, in turn, is equipped with a set of degrees of freedom. It is the task of the sensor control on level 0 to determine the concrete sensor parameters appropriate for carrying out commands submitted by sensor scheduling, in order to optimize the performance of the overall system.

The possible layers of sensor management discussed above are illustrated in Fig. 4, where typical issues are marked beside their corresponding levels. It should be noted, however, that the purpose of this discussion on levels of sensor management is intended only for better understanding and more convenient description of the general policy of top-down problem solving by a sensor manager. It is not the intention of this paper to give a generic prescription for system design, recognizing that functional requirements on sensor management are always problem dependent.

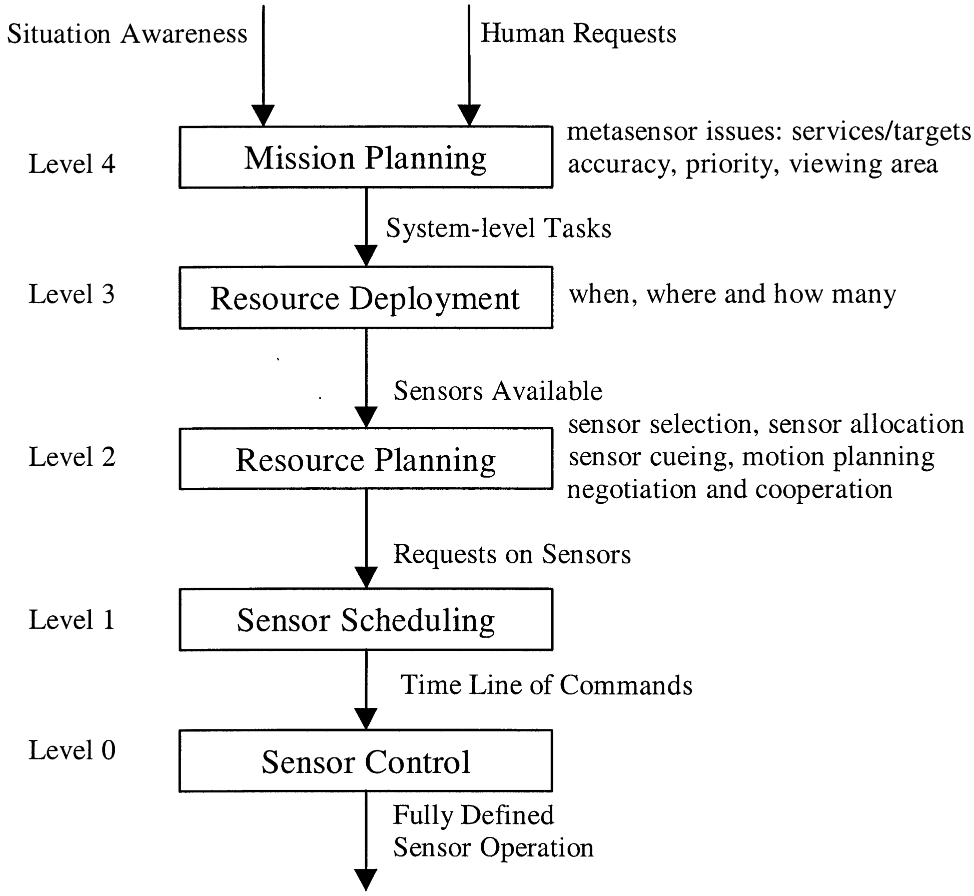

Fig. 4. Top-down problem solving by a sensor manager

We note that the above five-layered sensor management procedure does not exclude the use of feedback mechanisms. Indeed, situation awareness explicitly delivered to level four represents all feedback information from data fusion. Although not directly indicated in Fig. 4, we understand that required fusion results should be forwarded to other levels of the top-down procedure to support control and decision making therein.

This top-down problem solving procedure for sensor management is a kind of process model, describing the sequence of steps to be performed. Therefore it has only loose correspondence with the JDL function model. A set of resource management levels that more closely correspond to the JDL data fusion levels has been presented in [67], as an extension of the data fusion/resource management duality [11].

3 Principles of Sensor Placement

3.1 General Description

As stated in the problem description in section 1.4.1, sensor placement is viewed as a parameter optimization problem possibly subject to constraints. Two basic points are of concern here, namely the goal function for placement and the strategy to find optimal or near-optimal solutions. In order to keep pace with environment changes, e.g., the movements of targets, the optimization algorithms adopted must not impose a high computational load. Three optimization algorithms that have been employed for real-time sensor placement are *gradient descent* used in [57-58], *greedy local search* in [77], and *simulated annealing* in [59]. Worth noting is the fact that both greedy search and simulated annealing have the sometimes important merit of requiring no derivatives of the objective function in their search procedures. Various optimization criteria for sensor placement have been suggested. Generally they can be classified into *short-term* and *long-term* strategies depending on whether future performance is taken into account in the objective function.

Short-term sensor placement seeks to optimize some performance measure of the fusion system at the current stage but does not consider its future behavior. In view of this, sensor positions are determined simply according to one of the following criteria:

- Maximizing the probability of detection of the target [57-58],
- Minimizing the standard deviation in the target estimates [58],
- Maximizing the figure of merit for the sensor net [59].

Long-term sensor placement seeks to prolong the usage of deployed sensor(s) by taking care of possible future paths. Such an attempt complies with the general principle that sensor management should "value long-term goals of survival and success, not just accuracy and identity" [54]. Two examples of long-term placement are described in ([34-35], [71]) and [77] respectively. They both aim to find sensor locations such that the expected variance of the target state will be kept below a prescribed limit for as many consecutive time steps as possible after the present one. However, they differ in the state estimation methods used. The work by the FOI group applies Kalman filtering to identify target states so that it is computationally more efficient. On the other hand, the method by Williams is more generally applicable due to the employment of particle filtering methodology.

It bears mentioning that sensor placement in this context only refers to the issue of determining the positions where sensors are initially located. It does not involve planning the movements of mobile sensors after their deployment, which we believe is a kind of sensor behavior assignment rather than sensor deployment. By the same token, motion planning for movable sensors should be a function belonging to level two (resource planning) of the top-down procedure for sensor management proposed in section 2.2. This is an under-addressed topic but challenging for future research.

3.2 Signal Filtering Related to Sensor Placement

In target tracking applications, the sensor placement algorithm is inherently related to the filtering algorithm used for state estimation. The most commonly used is the Kalman filter, which assumes linear system models with additive Gaussian noise. A more recently investigated alternative is the particle filter [27] employed by ([58], [77]) for sensor placement in submarine tracking. The latter has the advantage of being able to handle any functional non-linearity and any distribution of system or measurement noise. Despite seeming quite different, both filtering approaches can be formulated within the general framework of recursive Bayesian estimation.

The Bayesian signal filter consists of essentially two stages: *prediction* and *update*, in order to construct the probability density function (pdf) of the state based on all available information. The prediction stage uses the system model to make prediction of the state at the next time step, and later the predicted state pdf is modified in the update stage using the newly acquired data from measurement. The update operation is an application of Bayes' theorem, which provides a mechanism to update prior knowledge in light of new evidence.

The goal is to derive the pdf of the current state x_k given the full set of measurements $D_k=\{y_1,\ldots,y_k\}$. Suppose the required pdf $p(x_{k-1}|D_{k-1})$ of the preceding time step k-1 is available. Then depending on the system model the prior pdf of the state at time k can be obtained by:

$$p(x_k \mid D_{k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid D_{k-1}) dx_{k-1} \tag{1}$$

where $p(x_k|x_{k-1})$ is the probabilistic model of the state evolution, defined by the system equation and known statistics of the system noise. Given a measurement at time k, the predicted prior pdf may be updated via Bayesian rule as:

$$p(x_k \mid D_k) = \frac{p(y_k \mid x_k) p(x_k \mid D_{k-1})}{\int p(y_k \mid x_k) p(x_k \mid D_{k-1}) dx_k} \tag{2}$$

Likewise $p(y_k \mid x_k)$ is another probabilistic model defined by the measurement equation and the known statistics of the measurement noise.

The recurrence relations (1) and (2) constitute the formal solution to the recursive Bayesian estimation problem. Unfortunately, analytic expressions of concrete results are only available under certain restrictions of the system and measurement models. For the linear-Gaussian estimation problems, the required pdf remains Gaussian at each iteration of the filter, and the Kalman filter can be used as a rigorous, explicitly formulated solution to the equations (1) and (2) by propagating and updating the mean and variance of the state distribution. In other cases with non-linear or non-Gaussian nature, however, there is no general analytic (closed form) expression available for the required pdf.

Particle filtering is indeed a functional approximate implementation of the general recursive Bayesian filter utilizing a random sample based representation of the state pdf. Given a set of independent random samples $\{x_{k-1}(i): i=1,\ldots,N\}$ drawn from the pdf $p(x_{k-1} \mid D_{k-1})$, the particle filter is a mechanism of propagating and updating these samples to get a set of new values $\{x_k(i): i = 1,\ldots,N\}$ which are approximately distributed as independent random samples of $p(x_k \mid D_k)$. In view of this, the equations of prediction and update in the original Bayesian filter need to be approximated accordingly to comply with the sample representation requirement. The basic strategies for prediction and update used in the standard form of particle filtering are given in the following.

Prediction: Considering the prior pdf $p(x_k \mid D_{k-1})$ to be approximated as

$$\begin{aligned} p(x_k \mid D_{k-1}) &= \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid D_{k-1}) dx_{k-1} \\ &\approx N^{-1} \sum_{i=1}^{N} p(x_k \mid x_{k-1} = x_{k-1}(i)) \end{aligned} \tag{3}$$

its required sample set $\{x_k^*(i): i = 1,\cdots,N^*\}$ can be generated by repeating N^* times of the procedure below:

(a) uniformly resample with replacement from the set of values $\{x_{k-1}(i): i = 1,\ldots,N\}$

(b) pass the resampled value through the system model to produce a sample $x_k^*(i)$ from the prior at time step k

Note that N^* is not necessarily greater than N.

Update: On receipt of the measurement y_k, evaluate a normalized weight, q_i, for each sample from the prior using the measurement model:

$$q_i = \frac{p(y_k | x_k^*(i))}{\sum_{j=1}^{N^*} p(y_k | x_k^*(j))} \tag{4}$$

This defines a discrete distribution over $\{x_k^*(i): i = 1,\cdots,N^*\}$ with probability mass q_i associated with sample $x_k^*(i)$. Now resample N times from the discrete distribution to produce samples $\{x_k(i): i = 1,\cdots,N\}$ such that $Prob\{x_k(j) = x_k^*(i)\} = q_i$. In this way we obtain a set of samples that tend in distribution to the posterior pdf $p(x_k \mid D_k)$ as N tends to infinity. The justification for the update phase of the particle filter was given in [64], where it was proved that the Bayesian theorem can be implemented as a weighted bootstrap.

A major drawback associated with the particle filter is its heavy computational demand due to the necessity of simulating the behavior of a large cloud of particles. Only recently with the availability of high speed computers have applications of this methodology become possible. On the other hand, many techniques have been proposed to improve the computational performance and/or accuracy of the particle filter method, one interesting among them being the Rao-Blackwellised version [20].

4 Observation Task Evaluation

Specification of importance (priorities) of different observation tasks, based on the current tactical situation, is significant to adapt sensor operations to changing environments. Information about task priorities can be very useful in scheduling sensor actions and for negotiation among sensors in a decentralized architecture.

4.1 Evaluation Based on Fuzzy Decision Trees

Molina Lopez et al [53] developed a symbolic reasoning process to infer numeric evaluations of defense surveillance tasks using fuzzy decision trees. Each node in the decision tree is a linguistic variable with its possible values being fuzzy subsets in the universe of discourse for its corresponding concept, and the relations between nodes are defined by fuzzy if-then rules. Starting from information provided by situation assessment and/or other data fusion processes, the inference engine proceeds through the tree by generating intermediate conclusions, until the task priority associated with the root node has been drawn. A tool based on parser generator technology was described in [69] for creating a fuzzy decision tree from rules (or grammar clauses).

A decision tree presented in [53] for search task priority is redrawn in Fig. 5 with small simplification. The root node of this tree is *search task priority*, which is directly affected by the nodes: *Vulnerability of Sector*, *Danger of Sector*, and *Targets in the Last Search*. Further, among the three concepts directly related to the root node, there are two intermediate conclusions (*vulnerability* and *danger*) whose values are to be deduced from leaf nodes.

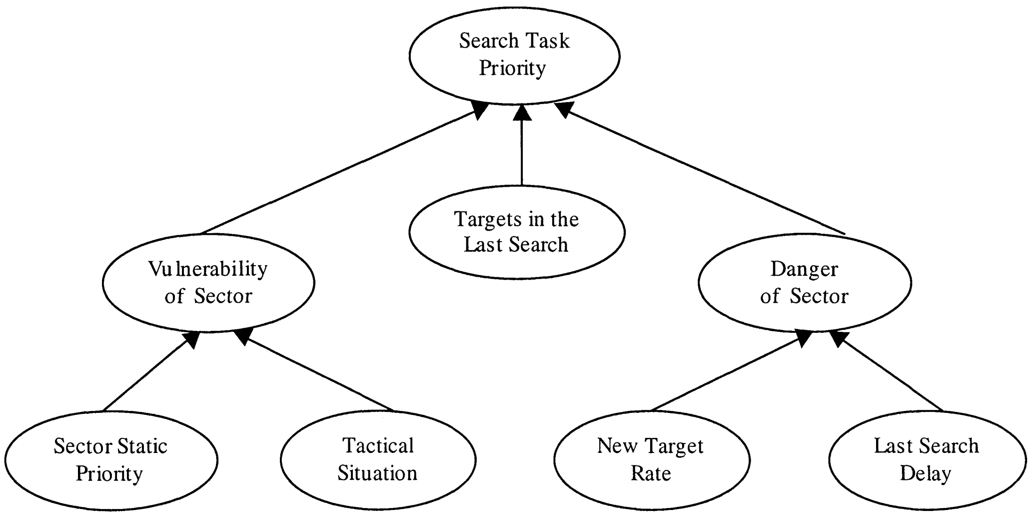

Fig. 5. A decision tree to determine the value of search task priority (modified from [53])

Five fuzzy labels: *very low*, *low, medium, high*, and *very high* are used to characterize *Vulnerability of Sector* and *Danger of Sector*, while the concept *Targets in the Last Search* corresponds to another three fuzzy labels: *normal*, *many*, and *too many*. Taking into account every AND combination of these fuzzy labels, a rule set consisting of 5×5×3=75 linguistic rules has been available to describe the dependence of the search task priority on the three nodes directly related to it. A rule in this rule base is of the type:

R_i: IF (*vulnerability* = A_{i1}) and (*danger* = A_{i2}) and (*targets* = A_{i3}) THEN (Priority = B_i)
(i = 1, 2, …, 75)

In the above example, A_{i1}, A_{i2} are labels from {*very low*, *low*, *medium*, *high*, *very high*} and A_{i3} one label from {*normal*, *many*, *too many*}, and B_i is a fuzzy set for search task priority with its membership function $B_i(y)$ defined on the domain of priority degrees Y. Under the Mamdani model of fuzzy reasoning the output fuzzy set F_i, inferred by the ith rule is

$$F_i(y) = \tau_i \wedge B_i(y) \tag{5}$$

where τ_i denotes the firing strength of the *ith* rule, defined for given crisp values of vulnerability, danger, and targets by

$$\tau_i = A_{i1}(vulnerability) \wedge A_{i2}(danger) \wedge A_{i3}(targets) \tag{6}$$

The output fuzzy sets F_i, inferred by the individual rules are then aggregated by an OR aggregating operator resulting in an overall output fuzzy set F for search task priority, where

$$F(y) = \bigvee_{i=1}^{75} F_i(y) = \bigvee_{i=1}^{75} [\tau_i \wedge B_i(y)] \tag{7}$$

Finally, the crisp value of search task priority is obtained by defuzzification of the output fuzzy set F, deriving a crisp representation of it. Usually we have two methods available for implementing this purpose. The first is calculation with the *Center of Area* (*COA*) method, i.e.,

$$y^{COA} = \int_Y yF(y)dy \Big/ \int_Y F(y)dy \tag{8}$$

The other alternative is to define the value of priority as the *Mean of Maxima* of the membership function $F(y)$. That is

$$y^{MOM} = \sum_{y_j \in G} y_j / Card(G) \tag{9}$$

where G is a subset of Y consisting of those elements in the domain which have maximum value of $F(y)$ thus

$$G = \{ y \in Y \mid F(y) = \max_y F(y) \} \tag{10}$$

At this point we have outlined all the necessary steps to derive the priority value based on the three nodes directly related to the root node. The intermediate conclusions like *Vulnerability of Sector* and *Danger of Sector* in the decision tree can be inferred from their respective leaf nodes using the same procedure as above.

4.2 Evaluation Using Neural Networks

Komorniczak et al [40] proposed to assign target priorities using a neural network, which accepts features of detected targets as its inputs and offers importance levels of these targets as its outputs. Given sufficient learning examples, the neural network can be built via off-line training based on a learning algorithm such as back propagation. The authors utilized a rather simple neural network for this purpose as shown in Fig. 6, where each input is multiplied with the corresponding weight W_i and these products are summed up yielding a net linear output u. The activation function to compute target importance levels is of the form:

$$f(u) = 1/(1 + \exp(-bu)) \tag{11}$$

with b being a parameter determining the slope of the function.

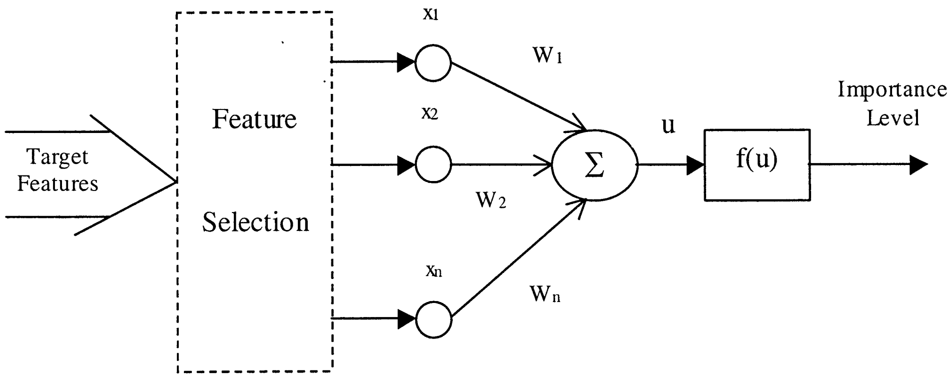

Fig. 6. Neural network for (target) priority assignment

However, to create the required training set, an experienced human operator is needed to evaluate beforehand target importance levels with respect to a large number of various situations. An issue could be whether it is possible or convenient to collect sufficient examples for the network training.

In addition, in our view, identifying significant features as inputs before neural network modeling could be advantageous in order to filter out irrelevant features, alleviating computational demands and reducing the risk of overfitting in data-based training. This motivates including the dashed block *feature selection* in Fig. 6 as a potentially desirable component, although this issue was not addressed in the cited paper.

4.3 Evaluation Based on Goal Lattices

A goal-lattice-based methodology was addressed in ([32], [48]) with the purpose of quantifying the relative contribution of actual observation tasks in the context of a complex mission. In doing this, one has to identify all goals relevant to that mission and then define an ordering relation among them, leading to the construction of a partially ordered set. The ordering relationship used is a precedence ordering induced by the simple statement "this goal is necessary in order for the other goal to be satisfied". The established partially ordered set can be represented as a lattice, allowing for straightforward apportionment of goal values from higher relatively abstract goals to lower increasingly concrete goals. At each layer of the lattice, each goal gets a value as the sum of the values received from the (higher) goals from which it is included. On the other hand, a goal also distributes its value among the (lower) goals that it includes. In particular, the bottom layer of the lattice comprises the nodes for actual observation tasks with the apportioned values reflecting their relative importance to accomplish the mission. A web browser implementation of a goal lattice constructor has recently been reported in [30].

An illustrative example was presented in [48] applying the goal lattice approach to an Air Force mission in an "in harm's way" scenario. Seventeen goals relevant to the "in harm's way" situation were identified with associated interrelations among them as shown in Table I. The resulting goal lattice is depicted in Fig. 7, where the value of each goal is uniformly distributed among its included goals presuming that all included goals contribute equally to their including goal. The bottom layer corresponds to three fundamental observation tasks: tracking, identifying, and searching with importance degrees of 0.36, 0.46, and 0.18 respectively. As a further aside, the 17 goals listed in Table I constitute a subset of applicable Air Force Doctrine goals. In publications of the United States Air Force, totally 90 Air Force mission goals were defined within the six mission areas Offensive Counterair, Defensive Counterair, Air Interdiction, Battlefield Air Interdiction, Close Air Support, and Suppression of Enemy Air Defenses. A global goal lattice for the whole set of mission goals was also established in [48].

Table I: "In Harm's Way" Goals

Goal Number	Goal	Included Goals
1	to obtain and maintain air superiority	2, 3, 4, 5
2	to minimize losses	6, 7, 8
3	to minimize personnel losses	6, 7, 8
4	to minimize weapons expenditure	6, 8
5	to seize the element of surprise	8
6	to avoid own detection	9, 10
7	to minimize fuel usage	10, 11
8	to minimize the uncertainty about the environment	12, 13
9	to navigate	15, 16
10	to avoid threats	15, 16
11	to route plan	15, 17
12	to maintain currency of the enemy order of battle	14, 16
13	to assess state of the enemy's readiness	14
14	to collect intelligence	15, 16, 17
15	to track all detected targets	
16	to identify targets	
17	to search for enemy targets	

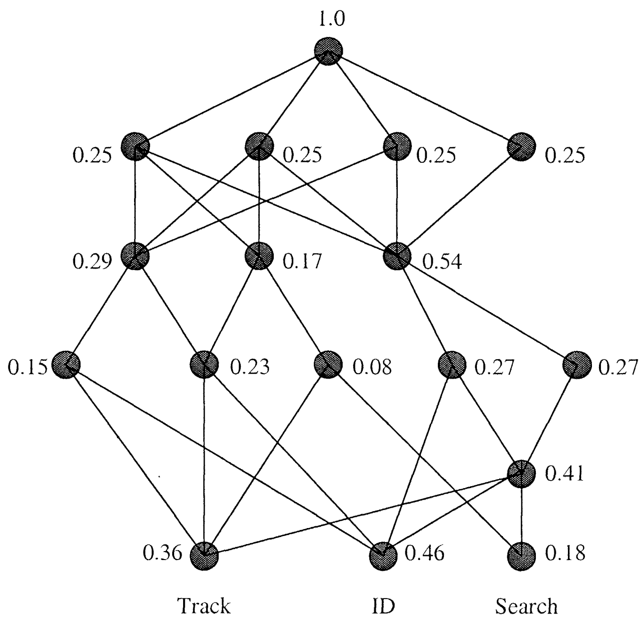

Fig. 7. Lattice of the "In Harm's Way Goals" (cited from [48])

One major difficulty involved with the goal lattice approach lies in the distribution of goal values. A uniform distribution as shown in Fig. 7 is easy to implement but may be a gross distortion of the true nature of the underlying problem, since it is likely that some included goals contribute more or are more important for the accomplishment of the including goal than others. Weights of arcs in the lattice reflect the preferences of some goals over others and these preferences can change between different phases of a mission. It is doubtful whether it is realistic for an operator to define an exact distribution function for every goal and in every mission stage, particularly in real-time. Perhaps (fuzzy) expert system technology could be incorporated to automate the process of determining arc weights during a mission, in response to changes of information provided by data fusion. However, the issue of how to define distribution functions for individual goals was not addressed by the authors.

5 Measurement Policies for Information Collection

What measurement policies to follow for the collection of adequate information is an issue which needs to be taken into account by the sensor manager, involving the type (pattern) of measurement to be used, the frequency of observation on targets, the strategy of target detection by sensors, etc. It was suggested in [31] that the proper measurement type may be determined on the basis of information requests posed by mission goals or human operators. The term *information instantiation* was used to signify a collection of methods, which map information needs to measurement functions. The problem of sensor-target detection is another significant topic related to measurement policies in the context of target searching. Inherently, a strategy to detect an unknown target can be viewed as a measurement sequencing process under uncertainty, which has been addressed by some researchers using stochastic dynamic programming and reinforcement learning (RL) techniques, see ([8], [14], [47]). In the following we will detail sensor measurement policies in terms of *measurement pattern*, *target observation frequency*, and *target detection strategy* respectively.

5.1 Which Measurement Pattern to Use

In the case of search, the characteristics of various measurement patterns include high- or low-bearing resolution, high- or low-range resolution, direct velocity measurement (Doppler), or any combination of these capabilities. A general idea proposed by [31] is to determine the measurement pattern in two steps. The first step is to downselect from all available measurement options that can be performed by the sensor system to the admissible ones, subject to constraints. Then a local optimization criterion can be applied in the second step to make a choice from the set of admissible solutions, maximizing the incidental information value of the pattern to be performed.

In response to requests of object identification, table look-up was suggested in [31] for deciding what pattern of measurement will enable the desired identification. For instance, if only the type of target is needed, it is sufficient to use electronic support measures (ESM) to observe signals emanating from the platform and then make classification of the target in terms of the electronic order of battle (EOB). Facing a more detailed task of hull-to-emitter correlation, however, we would have to arrange a longer observation period due to certain characteristics of ESM sensors. Such prior knowledge along with numerous identification techniques, their applicability, and operational constraints have to be stored in a predetermined list for real-time usage during decision making.

5.2 When to Observe a Tracked Target

When to make observations is an important issue affecting the accuracy of target tracking. Since the uncertainty of the target position is increasing in the absence of measurements, one has to update the track before the development of uncertainty exceeds an acceptance limit. Assuming the use of a Kalman filter as the tool for state estimation, the error covariance matrix can be adopted as an indicator of the state variance. The problem is therefore to find out the latest time for the next measurement, which still results in an updated covariance matrix falling within the specified constraint. According to [31], the maximum number of intervals between two successive measurements can be derived as follows:

The starting point of the analysis is related to the error covariance prediction equation

$$P(k \mid k-1) = F(k-1)P(k-1 \mid k-1)F(k-1)^{\mathrm{T}} + Q(k-1) \tag{12}$$

and the error covariance update equation

$$P(k|k) = \left[I - K(k)H(k)\right] \cdot P(k|k-1) \tag{13}$$

where

$$K(k) = P(k|k-1)H(k)^{\mathrm{T}}\left[H(k)P(k|k-1)H(k)^{\mathrm{T}} + R(k)\right]^{-1} \tag{14}$$

with $F(k)$, $H(k)$ denoting the state evolving matrix and the system observation matrix respectively. $Q(k)$ is the system noise covariance and $R(k)$ the measurement noise covariance.

Providing no measurement was performed at time k-1, the observation matrix $H(k$-1) becomes null so that we have

$$P(k-1 \mid k-1) = P(k-1 \mid k-2) \tag{15}$$

and $$P(k \mid k-1) = F(k-1)P(k-1 \mid k-2)F(k-1)^{\mathrm{T}} + Q(k-1) \tag{16}$$

Similarly to equation (12), the predicted error covariance at time k-1 can be written as

$$P(k-1 \mid k-2) = F(k-2)P(k-2 \mid k-2)F(k-2)^{\mathrm{T}} + Q(k-2) \tag{17}$$

Substituting (17) into (16) results in

$$\begin{aligned} P(k \mid k-1) = {} & F(k-1)F(k-2)P(k-2 \mid k-2)F(k-2)^{\mathrm{T}}F(k-1)^{\mathrm{T}} + \\ & F(k-1)Q(k-2)F(k-1)^{\mathrm{T}} + Q(k-1) \end{aligned} \tag{18}$$

Supposing now that n_t intervals have been skipped without observation until the time step k, we can reprocess equation (18) backwards to step k-n_t, getting the following recursive expression:

$$\begin{aligned} P(k \mid k-1) = {} & \left(\prod_{j=1}^{n_t} F(k-j)\right) P(k-n_t \mid k-n_t) \left(\prod_{j=1}^{n_t} F(k-n_t-j+1)^{\mathrm{T}}\right) + \\ & \sum_{j=1}^{n_t-1}\left(\left(\prod_{m=1}^{n_t-j} F(k-m)\right) Q(k-n_t+j-1)\left(\prod_{m=1}^{n_t-j} F(k-n_t+j+m-1)^{\mathrm{T}}\right)\right) + Q(k-1) \end{aligned} \tag{19}$$

Given that the upper limit of the updated error covariance is known, the allowed maximum covariance before update is derived from

$$P(k \mid k)^{-1} = P(k \mid k-1)^{-1} + H(k)^{\mathrm{T}} R(k)^{-1} H(k) \tag{20}$$

Finally, having obtained the allowed maximum $P(k \mid k-1)$ and the last updated error covariance matrix $P(k-n_t \mid k-n_t)$, we can acquire the maximum value of n_t from equation (19).

In our opinion, identification of the maximum time between successive track updates while maintaining the desired uncertainty level is the most valuable contribution of the paper [31]. The connotation is that observation of the target at every time step is not a compelling demand. The question of when it is necessary to make the next measurement is answered by equation (19). This provides a rigorous and useful guidance for reduction of sensor resources usage, which is of particular significance for target rich but sensor poor applications.

5.3 In What Sequence to Detect an Unknown Target

In the target detection problem, one maintains a vector of probability estimates: $[\pi_1(t), \pi_2(t), \cdots, \pi_k(t), \cdots]$ for all cells in the frame, with $\pi_k(t)$ denoting the conditional probability of a target being located in cell k given given all the previous measurements made in that cell. At every stage t, a noisy measurement is made on a certain cell k reporting whether detection was made or not, and the prior probability estimate $\pi_k(t-1)$ for cell k is thereby updated based on Bayes' theorem by

$$\pi_k(t) = \begin{cases} \dfrac{(1-P_{MD})\cdot\pi_k(t-1)}{(1-P_{MD})\cdot\pi_k(t-1)+P_{FA}\cdot(1-\pi_k(t-1))}, & \text{if reporting detection} \\ \dfrac{P_{MD}\cdot\pi_k(t-1)}{P_{MD}\cdot\pi_k(t-1)+(1-P_{FA})\cdot(1-\pi_k(t-1))}, & \text{if reporting no detection} \end{cases} \tag{21}$$

where P_{FA} = P(false alarm) = P(detect | absent)
P_{MD} = P(missed detection) = P(not detect | present)

Combined with this problem we need a target detection strategy, which determines the sequence of cells to measure such that the sensor(s) can focus on a proper area in every stage to get the best final result. So far the following three strategies have been available to suit this purpose:

1) *Direct Detection.* This is an uninformed detection pattern, which advances through the whole frame in a predetermined cell sequence. Upon completing the frame, the procedure of measurements is repeated again in the same order as the previous one. In order to ensure that equal attention is dedicated to each cell, the total number of measurements is usually chosen to be a multiple of the number of cells in the frame so that incomplete frames of measurements are obviated. Indeed, because of its low efficiency this mode of detection is only used as a default mode in some field systems.

2) *Index Rule Detection.* This strategy chooses the most likely cell (i.e. the cell with the highest probability estimate) to be measured at every stage. At the beginning, all cells are assumed to be equally probable, therefore a random cell is selected for measurement as an initialization of the detection procedure. Later with probability updates per equation (21), measurements are expected to gradually congregate on the target cell whose probability estimate becomes dominating over others', due to repeated sensor reports of target detection there. Castanon [14] analyzed this detection strategy using concepts from stochastic dynamic programming and concluded that the probability to find the target is maximized by a simple index rule under the condition of symmetric measurement densities. Since available

information is made use of by the index rule, the time required to correctly locate a target is shortened in comparison with direct detection.

3) *RL-Detection.* As a machine learning technique, reinforcement learning (RL) attempts to arrive at optimal system performance by trial and error through interactions with the environment. Certainly there may be several possibilities of using RL techniques to learn to conduct target detection tasks. The scheme proposed in [47] is to employ an RL-detection network for each cell to evaluate the reward of making a measurement there. The inputs to the one-cell network include the probability estimate $\pi_k(t)$ for the cell and the number of measurements that have been performed therein. Based on the network's outputs this strategy selects the cell with the largest predicted reward to receive the next measurement. At the beginning, the network's weights are set randomly, so that their predicted reward values are incorrect. But upon taking a new measurement, one can acquire an updated probability distribution of the target location using the Bayesian evidence rule and then calculate the expected reward value of this sensory action in terms of a predefined metric. The difference between the expected reward value and that, which is predicted by the network, provides useful information for correcting the weights of the network by means of a backpropagation algorithm.

6 Sensor Resource Allocation

Given that the tasks of observation have been postulated at the system level, the sensor manager should be able to distribute available sensing resources across them to best satisfy the information requests as a whole. Frequently, an observation task cannot be accomplished by a single sensor and thus a sensor combination must be applied to it, giving rise to the sensor selection problem. In other cases like multiple object classification, we may need dynamic allocation of sensors to achieve best identity accuracy after a certain number of measurement stages. In the sequel, existing approaches to this subject are summarized from a viewpoint of methodology.

6.1 Search-Based Approaches to Sensor Selection

The selection of a sensor subset to be applied to a given task can be considered as a search problem in combinatorial space. The goal is to find the most appropriate solution among all possible sensor combinations. Hence a key point involved is how to evaluate trials in the problem space.

Sensors' suitability was assessed in [25] using fuzzy set theory to establish a sensor preference graph, based on which the best sensor subset for a given task can be explored. Molina Lopez et al [52] introduced some heuristics to be associated with nodes of the search tree for sensor subset evaluation in terms of sensor load and sensor suitability. The *sensor load* constitutes an estimate of the sensor resources required to accomplish a task, while the sensor's *suitability* to perform a task is defined as a function of the necessity of fulfilling the task and the ability of the sensor to perform it. The objective function for a search algorithm should be constructed in a proper manner to reflect a good balance between the two factors.

In multi-sensor multi-target tracking applications, the problem of sensor selection simply means to decide suitable sensor combinations to be applied to measurement of different targets. Application of all sensors to all targets is usually not possible due to constraints in sensing and/or computational resources. Kalandros et al [37] proposed the so-called covariance control strategy for the purpose of maintaining a desired covariance level on each target while reducing system resource requirements. The covariance controller, as shown in Fig. 8, is responsible for determining a sensor combination for each target to meet the desired covariance level, while the sensor scheduler prioritizes the sensing requests from the covariance controller and issues commands regarding which actions are to be executed during every scanning interval.

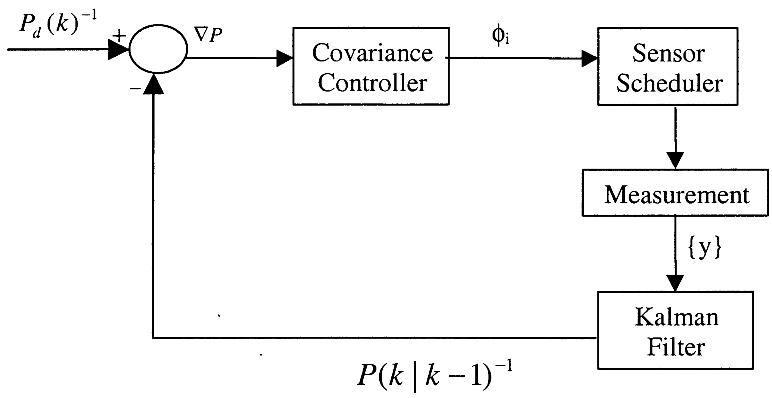

Fig. 8. Covariance control for sensor selection (modified from [37])

Assume that the target and its measurements can be described by the equations as:

$$x(k) = F(k-1)x(k-1) + w(k-1) \tag{22}$$

$$y_j(k) = H_j(k)x(k) + v_j(k) \tag{23}$$

In the above expression $x(k)$ is the current state of the target; $F(k)$, $H_j(k)$ are system matrices; and $y_j(k)$ denotes the measurements of the target from the *jth* sensor in the sensor combination $\varnothing_i$. $w(k)$ and $v_j(k)$ represent the system noise and measurement noise respectively, both of which are assumed to have zero-mean, white, Gaussian probability distributions. Based on the above assumptions, the sequential Kalman filter performs a separate filtering for each sensor in the combination and then propagates its estimate to the next filter. Therefore we write:

$$\begin{aligned} &\hat{x}_1(k\mid k) = F(k-1)\hat{x}(k-1\mid k-1) + K_1(k)(y_1(k) - H_1(k)F(k-1)\hat{x}(k-1\mid k-1)) \\ &\hat{x}_j(k\mid k) = \hat{x}_{j-1}(k\mid k) + K_j(k)(y_j(k) - H_j(k)\hat{x}_{j-1}(k\mid k)) \qquad j = 2,\cdots,\|\varnothing_i\| \\ &\hat{x}(k\mid k) = \hat{x}_{\|\Phi_i\|}(k\mid k) \end{aligned} \tag{24}$$

where

$$P(k\mid k-1) = F(k-1)P(k-1\mid k-1)F(k-1)^{\mathrm{T}} + Q(k-1)$$

and

$$\begin{aligned} &K_1(k) = P(k\mid k-1)H_1(k)^{\mathrm{T}}\left[H_1(k)P(k\mid k-1)H_1(k)^{\mathrm{T}} + R_1(k)\right]^{-1} \\ &K_j(k) = P_{j-1}(k\mid k)H_j(k)^{\mathrm{T}}\left[H_j(k)P_{j-1}(k\mid k)H_j(k)^{\mathrm{T}} + R_j(k)\right]^{-1} \qquad j = 2,\cdots,\|\varnothing_i\| \end{aligned} \tag{25}$$

with $Q(k)$ and $R_j(k)$ denoting the system noise covariance and the measurement noise covariance of the *j*th sensor (in the sensor combination) respectively.

The updating of state covariance for each filter is performed by

$$P_1(k \mid k) = (I - K_1(k)H_1(k))P(k \mid k-1)$$

$$P_j(k \mid k) = (I - K_j(k)H_j(k))P_{j-1}(k \mid k) \qquad j = 2,\cdots,\|\varnothing_i\|$$

$$P(k \mid k) = P_{\|\Phi_i\|}(k \mid k) \tag{26}$$

Alternatively, the covariance of state estimates can be calculated in a single step [4] by:

$$P(k \mid k)^{-1} = P(k \mid k-1)^{-1} + \sum_{j=1}^{\|\varnothing_i\|} H_j(k)^{\mathrm{T}} R_j(k)^{-1} H_j(k) \tag{27}$$

Defining $J_i = \sum_{j=1}^{\|\varnothing_i\|} H_j(k)^{\mathrm{T}} R_j(k)^{-1} H_j(k)$ as the sensor information gain corresponding to $\varnothing_i$, the demand on covariance requires that the value of J_i be as close to ∇P as possible, where

$$\nabla P = P_d(k)^{-1} - P(k \mid k-1)^{-1} \tag{28}$$

with $P_d(k)$ being the desired covariance for the target. As the J_i matrix for each sensor combination can be computed offline and stored in a library, only one matrix inverse $P(k \mid k-1)^{-1}$ needs to be calculated in each scan. On the other hand, the number of sensors applied to a target should be kept small in order to avoid an excessive computational burden. Considering both factors concurrently, the following three optimization objectives were suggested in [37] for the covariance controller to assign a desirable sensor combination to a target:

1) Eigenvalue/Sensors: use the sensor combination with the smallest number of sensors while maintaining all eigenvalues of the matrix $J_i - \nabla P$ positive. Selecting the fewest possible sensors ensures elimination of redundant sensor allocations, which was stressed by [54] as one of the general principles of sensor management.

2) Matrix Norm: use the sensor combination, which minimizes the norm of the covariance error $P_d(k) - P(k \mid k)$. The argument for this optimality criterion is to consider positive eigenvalues in the covariance error as excess resources applied to a target and negative eigenvalues as insufficient resources applied to that target. However, finding the minimum norm of the covariance error does not guarantee that the resulting covariance $P(k \mid k)$ will always stay within the desired limit.

3) Norm/Sensors: use the sensor set with the fewest sensors while keeping the norm of the covariance error within a predefined boundary.

The preferred merits of the strategy of covariance control decompose the sensor allocation problem into independent sub-problems for individual targets, each dealing with target-specific covariance goals. If the desired covariance changes for a target, then only the sensor set for that target needs to be reassigned. However, such a strategy is based on the assumption that sequential Kalman filtering is used to identify the states of targets and is therefore not applicable to cases where other filtering techniques, such as particle filtering, are involved. Additionally, the sensing requests from the covariance controller occasionally have to be delayed by the sensor scheduler due to resource constraints. The influence of possible execution delays on tracking performance remains for further study.

Once the search objective for a given task has been established, the second important issue for sensor selection is to determine which search algorithm to employ. The simplest attempt, as practiced in [52], is to examine every sensor combination that is capable of performing a given task before deciding on the best one. This mechanism was termed *global search algorithm* in [38] and its computational complexity was given as $O(2^{N_s} n^3)$ for target tracking problems, where N_S denotes the total number of available sensors and n is the size of the state vector of the target. In order to avoid the prohibitively high computational burden associated with global search, Kalandros et al [38] discussed two heuristic search algorithms that can reduce the computational demand at the cost of choosing a non-optimal sensor combination:

1) Greedy search: pick the best sensor into the combination at each iteration, until no improvement of the evaluation value can be acquired. The computational complexity of this algorithm is at most $O(\frac{n^3}{2}(N_S^2 + N_S))$ for target tracking applications.

2) Randomization and super-heuristics: begin with a base sensor combination (usually obtained with a priori knowledge or heuristic methods) and then generate alternative solutions via random perturbations to the initial combination. As a descendant of the super-heuristics technique introduced in [42], this search algorithm requires random perturbations to be performed on a non-uniform distribution, so as to increase the chance of finding a near-optimal combination in a few trials. Heuristic information such as Frequency-of-Selection and the solution from the greedy algorithm can be used to bias the random search towards those sensor combinations that are more likely to be good enough candidates for a given task.

6.2 Information-Theoretic Approaches to Sensor Selection

The idea of using information theory in sensor management was first proposed in [33]. From the information-theoretic point of view, sensors are applied to observe the environment in order to increase the information (or reduce the uncertainty) about the state of the world. It is the task of the sensor management system to make a reasonable allocation of sensing assets among various tasks such that the greatest possible amount of information is obtained at every measurement opportunity. In surveillance scenarios, information is gained when localizing a target or increasing the accuracy of the state estimate of a target being tracked. Observation of the target by sensor(s) enables the probability density distribution of the target location estimate to be updated, reducing the uncertainty about the target location. To quantify the information gained (uncertainty reduced) about a target due to sensor observation, two measures are available, based on *Shannon's entropy* and *Kullback-Leibler's cross-entropy* respectively.

Information Gain Based on Shannon's Entropy. According to the information theory of Shannon, the entropy of a random process is computed by

$$H_x = -K\sum_i p(X_i)\log p(X_i) \quad \text{for the discrete case}$$
$$= -K\int p(X)\log p(X)dX \quad \text{for the continuous case} \tag{29}$$

with K being any positive constant and $p(\bullet)$ denoting the probability density (mass) function for the continuous (discrete) case. Particularly for the n-variate and normal distribution, Shannon's entropy (with K=1) becomes:

$$H_x = \frac{n}{2}\log(2\pi e) + \frac{1}{2}\log(|P|) \tag{30}$$

where $|P|$ is the determinant of the covariance matrix P.

Interpreting this entropy as a measure of uncertainty, information can be quantified as the difference of the entropy between two given probability distributions of a random process. In view of this, we express the information gain, I, achieved on a target per measurement as

$$I = H_{\text{before observation}} - H_{\text{after observation}} \tag{31}$$

Further under the assumption of a normal distribution of the target state, the information gain in equation (31) is established by:

$$I(P_2,P_1) = \frac{n}{2}\log(2\pi e) + \frac{1}{2}\log(|P_1|) - (\frac{n}{2}\log(2\pi e) + \frac{1}{2}\log(|P_2|))$$
$$= \frac{1}{2}\log\left(\frac{|P_1|}{|P_2|}\right) \tag{32}$$

where P_1, P_2 are the covariance matrices before and after the measurement respectively.

Discrimination Gain Based on Kullback-Leibler's Cross-Entropy. The discrimination between the target's predicted density before observation, $p_1(X)$, and the updated density after observation, $p_2(X)$, is defined in terms of Kullback-Leibler's cross-entropy as:

$$D(p_2,p_1) = \int p_2(X)\log(p_2(X)/p_1(X))dX \tag{33}$$

Suppose both $p_1(X)$ and $p_2(X)$ are Gaussian vectors with means M_1 and M_2, and covariance P_1 and P_2, the discrimination gain of $p_2(X)$ with respect to $p_1(X)$ is given by

$$D(p_2,p_1) = \frac{1}{2}tr\left[P_1^{-1}\left(P_2 - P_1 + (M_2 - M_1)(M_2 - M_1)^{\mathrm{T}}\right)\right] - \frac{1}{2}\log\frac{|P_2|}{|P_1|} \tag{34}$$

Using either of the information measures described above, we can now develop a control paradigm for sensor selection in a multiple target situation. The purpose is to make decisions from a global information perspective by maximizing the amount of information gained across all targets. Given the expected information/discrimination gain for every pair of sensor subset and target, this problem can be addressed using a linear programming formulation as follows:

Let the sensors be indexed from 1 to N_S and the targets from 1 to N_T. Each sensor k has its own tracking capacity λ_k, representing the maximum number of targets that can be sensed by the sensor at each scan. Denoting the information/discrimination gain by G_{ij} when sensor subset i ($i = 1 \ldots 2^{N_s} - 1$) is allocated to target j, then the sensor selection problem is given by

Maximize $$Gain = \sum_{i=1}^{2^{N_S}-1} \sum_{j=1}^{N_T} G_{ij} x_{ij} \tag{35}$$

subject to constraints: $$\sum_{i=1}^{2^{N_S}-1} x_{ij} \le 1 \qquad \text{for} \quad j = 1 \ldots N_T \tag{36}$$

$$\sum_{i \in J(k)} \sum_{j=1}^{N_T} x_{ij} \le \lambda_k \qquad \text{for} \quad k = 1 \ldots N_S \tag{37}$$

with $x_{ij} \in \{0, 1\}$ for all pairs i, j

where $J(k)$ is the integer set containing all the indices of the sensor subsets in which sensor k is included.

In the literature both information measures have been shown useful for sensor selection in multi-sensor and multi-target tracking applications. Shannon's entropy was adopted in ([49], [61]) where the expected information gain can be derived by extrapolating the Kalman filter state covariance and then calculating the updated covariance matrix after a measurement. On the other hand, Schmaedeke et al [62] and Dodin et al [19] applied Kullback-Leibler's cross-entropy to measure the discrimination of information in combination with the Interacting Multiple Model Kalman Filter. For the time being, however, we have no knowledge about the comparative performance of the two entropy measures in sensor management applications.

In comparison with the covariance control strategy, information-theoretic approaches provide a globally optimal solution obeying all resource constraints so that a later scheduling of sensing requests is not needed. On the other hand, a higher computational burden should be expected when using them, owing to the required calculation of a large amount of information/discrimination gains at every cycle.

6.3 Decision-Theoretic Approaches to Sensor Planning

Kristensen [41] treated the problem of choosing proper sensing actions from a family of candidates as one of decision-making, and developed a framework based on *Bayesian decision analysis* (BDA) for its solution. It was strongly suggested in his thesis that uncertainty has to be dealt with in sensor planning since the ultimate purpose of performing sensory actions is to reduce uncertainty about the external world. On the other hand, BDA offers a powerful mechanism for representing and reasoning under uncertainty. The theory of BDA was primarily applied in the area of economics to evaluate various actions, e.g., investments, against each other.

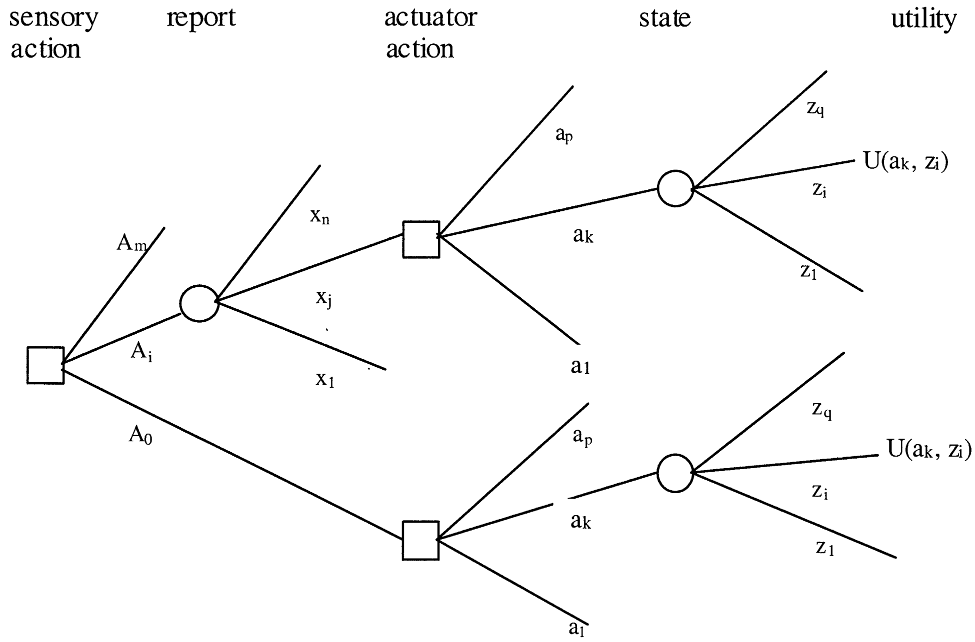

Fig. 9. Bayesian decision tree for sensor action planning (cited from [41])

Fig. 9 shows the decision tree which was used in [41] for sensor action planning, where we can see two different types of nodes: boxes and circles. A box is a *decision node* followed by paths decided by the system, while a circle corresponds to a *chance node* with its branches determined by chance. The root decision node is associated with sensory actions such as A_i which produces random output x_j related to the report chance node. The second decision node, called *actuator node*, represents the consequence type of action, e.g. a_k, leading to completion of the task. Furthermore, the state chance node corresponds to the world state after an actuator action, with $U(a_k, z_l)$ denoting the utility or payoff for completing a given task. From the viewpoint of buying information about an uncertain world state, BDA was used to evaluate distinct sensory actions against each other in a cost/benefit manner. To account for the increase in expected utility per cost unit compared with no sensing, the measure EISI (expected interest from sample information) was defined as:

$$EISI(A_i) = \frac{\sum_{j=1}^{n} P(x_j)EU(x_j) - EU(A_0)}{C(A_i)} \qquad (38)$$

where $EU(A_0)$, $EU(x_j)$ denote the expected utilities of no sensing and of receiving report x_j respectively, $P(x_j)$ the probability of receiving x_j and $C(A_i)$ the cost of performing the sensory action A_i. Finally, the Bayesian decision rule simply selects the one with the maximum EISI value as the sensory action to be performed. That is,

$$A_{opt} = \arg\max_{A_i = A_0}^{A_m} EISI(A_i) \qquad (39)$$

Although the method outlined above provides an explicit, coherent and theoretically well-founded framework to plan sensory actions under uncertainty, the construction of decision trees is still problem-dependent, in particular with respect to the subjective definition of utilities for completing a task. Whether it is possible to devise appropriate utilities in various applications remains for future investigation. The other potential difficulty that may occur is how to formulate actuator actions in the decision tree. It was noted in [41] that such actions are not restricted to operations of real actuators in the traditional sense but refer to somehow productive actions marking the completion of a task. However, we are not clear about what should be treated as actuator actions in accordance with the Bayesian decision tree when designing sensor managers for surveillance purposes.

Closely related to the work of Kristensen is the paper [43], which was based on the novel idea of learning the utility of using a sensor and which addressed the problem of how to find a set of sensors, which solves a given problem in cheapest possible fashion. Another decision-theoretic approach to so-called deliberate sensing was proposed in [21]. However, its authors did not give implementation details but pointed out that decision theory itself does not constitute a solution to the problem.

6.4 Fuzzy Logic Resource Manager

Gonsalves et al [26] described an *intelligent fusion and asset management processor* with its fourth level being a *fuzzy logic collection manager*, which is responsible for mapping the current situation state and enemy track information into (job) requests on sensing assets. The mapping depends on priori knowledge of sensor capabilities and enemy tactical doctrine. However, this paper does not explain how to incorporate fuzzy logic in the knowledge representation and reasoning.

A somewhat similar work to that of Gonsalves, called *fuzzy-logic based resource manager*, was presented in [65] which aims at optimal allocation of various resources distributed over many dissimilar platforms. A fuzzy decision tree for resource management was constructed via codifying related military expertise, with parameters of root concept membership functions being optimized by a genetic algorithm. The fitness function for the genetic algorithm was established based upon various considerations of geometry, physics, engineering and military doctrine. A later paper [66] by the same authors extends the idea of the former paper [65] by incorporating data mining techniques [6], which are generally understood as efficient discovery of valuable, implicit knowledge and information from a large collection of data, for performance optimization of the fuzzy decision tree. In particular, a genetic algorithm was applied in the data mining process to yield parameters of the root concept membership functions based upon a database of scenarios. Two possible ways to construct a database of good quality were discussed. A comprehensive scenario database is beneficial for the data mining technique to extract knowledge covering a wide range of behaviors and providing robust strategies to resource allocation. Otherwise a small data set containing a narrow spectrum of scenarios could result in a resource manager effective in some cases but ineffective in others.

6.5 Markov Decision for Sensor Allocation in Classification

For simultaneous classification of a number of unknown objects, it is important to properly distribute available sensors across different objects. The dynamic sensor allocation problem consists of selecting sensors of a multi-sensor system to be applied to various objects of

interest using feedback strategies. Mathematically this is a partially observed Markov decision problem ([44], [75]) as formulated in the following.
Consider a problem of N objects with $x_i \in \{1,\dots K\}$ denoting the true class of object i. It is assumed that x_i are modeled as independent random variables with discrete values in a finite space. Moreover each object i is associated with a prior probability distribution representing the a priori knowledge obtained on its class variable. Now in order to obtain further information about the state of each object, appropriate sensors should be assigned to various objects at the time intervals $t \in \{0,1, \dots, T\text{-}1\}$. The collection of sensors applied to object i during interval t is represented by a vector

$$U_i(t) = [u_{i1}(t) \ \cdots \ u_{iM}(t)] \tag{40}$$

where $$u_{ij}(t) = \begin{cases} 1 & \text{if sensor } j \text{ is used on object } i \text{ at interval } t \\ 0 & \text{otherwise} \end{cases} \tag{41}$$

and M is the total number of sensors in the sensing system. Because of limited resources sustaining the whole system, the planned sensor distributions must satisfy the following constraint for every $t \in \{0,1, \dots, T\text{-}1\}$

$$\sum_{i=1}^{N} \sum_{j=1}^{M} r_{ij}(t) u_{ij}(t) \le R(t) \tag{42}$$

where $R(t)$ is the maximum amount of resources that can be consumed by the whole sensor system during interval t and r_{ij} denotes that quantity of resources consumed by sensor j on object i.

Here, the goal of sensor allocation is to achieve an accurate classification of all the objects after T stages. Let $[v_1, v_2, \cdots, v_N]$ be the final classification decision and $c(x_i, v_i)$ the error function representing the penalty of classifying an object of x_i as type v_i. Sensor allocation can be defined as a problem to find a sequence of decisions $\{u_{ij}^*(t), v_i^* \mid i = 1 \cdots N, j = 1 \cdots M, t = 0 \cdots T-1\}$, which minimizes the expected total cost J_u in (43), subject to the constraint (42).

$$J_u = E\{\sum_{i=1}^{N} c_i(x_i, v_i)\} \tag{43}$$

The above problem is a special case of the finite-state, finite-observation, partially observed Markov decision problems addressed in ([44], [75]).

In principle, problems of this kind can be solved by means of *stochastic dynamic programming* (SDP). We can, in particular, convert the partially observed Markov decision problem into a standard fully observed one by defining the problem state in terms of its conditional probability distribution given the available information about the sensors applied and the measurement data acquired. Bertsekas [5] presented an SDP recursion to solve this problem. The main idea is to select prospective decisions at every stage in order that the expected value of the cost-to-go function at the next stage can be minimized. Unfortunately, the presence of the constraint (42) makes it impossible to decompose the whole task of decision making into smaller problems for individual objects. As a result, the dynamic programming algorithm must perform its job on the entire problem space and the associated computation becomes prohibitive even for a moderate number of objects.

The near-optimum feedback strategy proposed in [15] might possibly be useful to alleviate the computational burden associated with strict solutions to sensor allocation using SDP. By relaxing a hard constraint like that in (42) to an average resource constraint, the suggested approximate solution is based on the use of Lagrangian relaxation to decompose the multi-object optimization problem into many single object problems which are coordinated through introduced Lagrange multipliers. In this way, the relaxed sensor allocation problem is decoupled hierarchically into two levels. At the lower level, the sensors for single objects are determined using the SDP algorithm, under the given values of Lagrange multipliers. At the higher level, based on the lower level solutions, the values of Lagrange multipliers can be optimized via certain non-differentiable optimization techniques. The advantage of this approximate strategy is the reduction of computational complexity by decomposing the whole decision problem into decoupled sub-problems at the lower level. However, this benefit is earned at the cost of accepting a weak condition that the average amount of consumed resources does not exceed that which is available. It is controversial whether the proposal of relaxed resource constraint has practical value, since it allows violation of the original constraint in quite a few cases.

In addition to dynamic programming, state-based search methods were discussed in [10] for fully observable and non-observable Markov decision processes. In particular, it was pointed out that search techniques can be applied to partially observable problems as well, by searching the space of belief states [9], for example. The presented methodology of state-based search may be worth investigation in the context of sensor allocation, to achieve potentially more effective and efficient decisions about sensor distributions in real-time.

7 Towards Cooperative Sensor Behavior

A team-theoretic formulation of multi-sensor systems was presented in [29] with sensors being considered as members or agents of a team, able to offer opinions and to bargain in group decision situations. Coordination and control under this structure was analyzed and discussed based on a theory of *team decision-making*. It was assumed that a manager/coordinator makes group decisions according to criteria of maximum team utility. The most general form of team utility is to incorporate utility measures of both team member and joint team actions to combine local and global objectives. Alternatively, in some cases, the team utility function can disregard individual team member opinions, thereby allowing the manager to make optimum decisions regardless of individual team member losses. It is clear, however, that since a team utility represents group preference with respect to an underlying problem in a specific scenario, its definition is situation dependent. Thus, establishing a comprehensive function for assessing team utility may become difficult if not impossible in complex applications.

Bowyer et al [12] analyzed the issue of integrating distributed sensor assets based on the concept of cooperating machines. It was argued that heterogeneous sensors in a multi-sensor system should be treated as cooperative members of a *community*. Here, the concept of community refers to a plastic association rather than a rigid connection of sensors, considering the possible departure of a previously participating sensor as the situation unfolds. The authors also pointed out that certain social behavior principles are desired of a sensor network to foster cooperation between individual members and to enhance their mutual awareness. Such considerations led to the proposal of exploiting agent technology for managing temporal requirements and implementing the desired social behaviors within a community of sensors. A multi-agent based architecture was suggested for this purpose, allowing agents to interact with the sensor system. As indicated by the proposed agent system shown in Fig. 10, the information from an existing sensor is wrapped up by its local sensor agent, which then communicates and negotiates with other agents via a suitable communication medium.

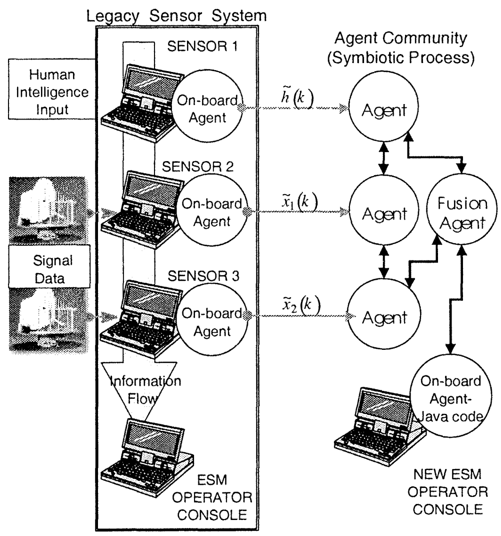

Fig. 10. A multi-agent based architecture for sensor integration (cited from [12])

A multi-agent resource management system was also studied by Luo et al [45] to support the integration of data from multiple sensors and associated important sensory information located in different agents. Two kinds of agents were involved in this study, with service agents handling local sensor information and a global decision agent conducting necessary coordination activities to achieve cooperative behavior of the whole system. An Internet-based network architecture was used for communications between service agents and the global decision agent.

Although approaches based on intelligent agents seem to offer a useful framework to realize autonomous cooperation between different sensors, few concrete policies have been reported in the literature answering the question how such agents could coordinate their decisions in a complex environment. However, for the simple problem of switching between sensors when tracking a moving target, a fuzzy controller has been designed for the cueing process [55] and shown to be very successful in simulations.

Currently, the Argus project [22] is carried out by the *Australian Centre for Field Robotics* in Sydney, to develop a decentralized ground-based sensor network. The nodes include vision, multi-spectral cameras, lasers, and (in the future) radars. The Argus network will be used for both air-target tracking and hand-over of ground targets from air to land sensors.

8 Conclusion and Discussion

This paper highlights the emerging area of multi-sensor management from the perspective of multi-sensor information fusion. Characteristics of three problem classes of multi-sensor management are elaborated. Furthermore, a general perspective on problem solving for sensor management is presented in terms of certain methodological considerations and a top-down working procedure. As a collection of activities controlling information gathering, the scope of sensor management is wide and many problems involved seem to be case-specific. In view of this, this paper discusses the state-of-art in terms of solutions to specific problems.

Research in sensor management is still in its infancy. Recent early work on this topic has been largely focused on sensor scheduling and sensor resource allocation. With the advances of information fusion systems into more and more complex environments, an increasing demand will likely occur for sensor management functions on high levels such as mission planning, sensor deployment and sensor coordination. Unfortunately, these issues have not yet been extensively addressed.

Contemplating sensor management requirements on high levels leaves us with the sense that methods of qualitative reasoning may be needed as complementary tools, to deal with abstract but imprecise information. In general, highly strategic decisions in sensor management might require the involvement of human operators, who are expected to "discuss" and "negotiate" with the software for working out intelligent solutions. Usually human operators tend to express their opinions in the form of propositions formulated in natural language and they will also want to receive information as linguistically understandable descriptions. Moreover, tolerance of imprecision seems a necessary requirement when making high-level decisions for achieving tractability, robustness, low computational cost, and better rapport with the inexact nature of human thinking.

We feel that the term sensor management is becoming inappropriate to capture what it really means to cope with increasingly high-level functionality in the future. This term was originally put forward to refer mainly to single sensor scheduling and control. It does not accurately capture the nature of process refinement in the context of information fusion. Indeed what we are now talking about is a procedure to manage data acquisition from the external world driven by information requests. Whether a new term is necessary or beneficial to better reflect the gist of this fast developing area is open for debate and discussion by the information fusion community.

Future research work can be conducted on several interesting but under-researched problems, including:

- Combined sensor placement and sensor selection in multi-sensor and multi-target applications. So far in the literature these two problems have been addressed in isolation of each other. Some papers discuss sensor selection for various targets under the assumption of known fixed sensor locations, whereas other papers focus on sensor placement in tracking a single target, so that sensor selection is not required. However, in real applications of multi-sensor and multi-target tracking we frequently have to solve these two problems in combination.

- Management of movable sensor assets or platforms such as unmanned air vehicles for surveillance purposes. An interesting problem may be planning time-constrained movements of sensor(s) to keep track of moving targets in a dynamic environment.

- Cooperation between sensors in a decentralized paradigm. Cooperation is particularly needed for a team of sensors and platforms that collectively perform surveillance tasks across a region. Each team member is expected to make its own decision about where to go and what to sense. But these decisions must be coordinated with other sensors and platforms to arrive at a globally optimal perception for the system as a whole. Further, in the case of failure of one or more sensors or platforms, the system should be able to reconfigure the control and coordination of sensor behaviors.

- Sensor management under multiple objectives. Prevalent in military applications is that there may be other objectives to be accounted for, such as remaining covert and recognizing enemy spoofing, in addition to information fusion quality. Sometimes such demands conflict with each other, requiring on one hand measuring in a highly covert way and on the other hand getting as much information as possible. In order to achieve a good balance between different objectives, an appropriate arbitration mechanism is needed.

- Adaptive signal filtering with respect to sensor placement. Up to now Kalman filtering and particle filtering are used in sensor placement for target state estimation. Both techniques have advantages and weaknesses. An adaptive mechanism is desirable to decide which filter to use based on the situation and the mission goal. Moreover, both Kalman and particle filters give average performance of state estimation. However, in some cases we may want to ensure the worst-case accuracy, especially in military applications. This could make it advantageous to use so-called $H\infty$ filtering (indeed Kalman filtering is also termed H_2 filtering). $H\infty$ filtering minimizes the "worst-case" estimation error and assumes no priori knowledge of noise statistics [63]. Automatic switching among the three filters could be valuable for implementation of adaptive signal filtering.

Acknowledgements. We would like to sincerely thank Johan Schubert for his helpful suggestions, Ronnie Johansson and Karsten Jöred for beneficial discussions and remarks.

References

[1] R.A. Adrian, Sensor management, in: Proceedings of the AIAA/IEEE Digital Avionics System Conference, Ft. Worth, TX, 1993, pp. 32-37.

[2] R.C. Arkin, Behavior-based robotics, MIT Press, Cambridge, MA, 1998.

[3] S. Arnborg, H. Artman, J. Brynielsson and K. Wallenius, Information Awareness in Command and Control: Precision, Quality, Utility, in: Proceedings of the International Conference on Information Fusion, 2000, pp. ThB1-25--32.

[4] Y. Bar-Shalom and X. Li, Estimation and tracking: Principles, Techniques, and Software, Artech House, Boston, 1993.

[5] D.P. Bertsekas, Dynamic programming and optimal control, Vols. I-II, Athena Scientific, Belmont, MA, 1995.

[6] J.P. Bigus, Data mining with neural nets, Chapter 1, McGraw-Hill, New York, 1996.

[7] S. Blackman, and R. Popoli, Sensor management, in: Design and analysis of modern tracking systems, Artech House, Norwood, MA, 1999, pp. 967-1068.

[8] E. Blasch, R. Malhotra and S. Musick, Using reinforcement learning for target search in sensor management, in: Proceedings of the National Sensor and Data Fusion Conference, Boston, MA, 1997, pp. 267-275.

[9] B. Bonet and H. Geffner, Learning sorting and decision trees with POMDPs, in: Proceedings of the 15th International Conference on Machine Learning, Madison, WI, 1998, pp. 73-81.

[10] C. Boutilier, T. Dean and S. Hanks, Decision-theoretic planning: structural assumptions and computational leverage, Journal of Artificial Intelligence Research 11 (1999) 1-94.

[11] C. Bowman and A. Steinberg, A systems engineering approach for implementing data fusion systems, In: D. Hall and J. Llinas, eds. Handbook of Data Fusion, Chapter 16, CRC Press, 2001, pp. 16-1--39.

[12] R.S. Bowyer and R.E. Bogner, Cooperative behavior in multi-sensor systems, in: Proceedings of the 6th International Conference on Neural Information Processing, Perth, Australia, 1999.

[13] R.A. Brooks, Intelligence without reason, in: Proceedings of the 12th International Joint Conference on Artificial Intelligence, Sydney, 1991, pp. 569-595.

[14] D.A. Castanon, Optimal detection strategies in dynamic hypothesis testing, IEEE Transactions on Systems, Man, and Cybernetics 25 (7) (1995) 1130-1138.

[15] D.A. Castanon, Approximate dynamic programming for sensor management, in: Proceedings of the 36th IEEE Conference on Decision and Control, 1997, pp. 1202-1207.

[16] C. Chong, S. Mori and K. Chang, Distributed multitarget multisensor tracking, in: B. Shalom, eds. Multitarget-multisensor tracking: applications and advances, Chapter 8, Artech House, Boston, 1990.

[17] M. Cone and S. Musick, Scheduling for sensor management using genetic search, in: Proceedings of the IEEE National Aerospace and Electronics Conference, Vol. 1, Dayton OH, 1996, pp. 150-156.

[18] B. Dasarathy, Decision Fusion, IEEE Computer Society Press, 1994.

[19] P. Dodin, J. Verliac and V. Nimier, Analysis of the multisensor multitarget tracking resource allocation problem, in: Proceedings of the International Conference on Information Fusion, 2000, pp. WeC1-17--22.

[20] A. Doucet, N. de Freitas, K. Murphy, S. Russell, Rao-Blackwellised particle filtering for dynamic Bayesian Networks, in: Proceedings of the 16th International Conference on Uncertainty in Artificial Intelligence, Stanford, 2000, pp. 176-183.

[21] D.L. Draper and S. Hanks, Sensing as a deliberate act. in: Working Notes from the AAAI Spring Symposium on Control of Selective Perception, Stanford, Cal., 1992, pp. 39-42.

[22] H. Durrant-Whyte, A beginner's guide to decentralised data fusion, Technical Document of Australian Centre for Field Robotics, University of Sydney, Australia, 2000.

[23] H. Durrant-Whyte and M. Stevens, Data fusion in decentralised sensing networks, in: Proceedings of the International Conference on Information Fusion, 2001, pp. WeA3-19--24.

[24] R. Fung, E. Horvitz and P. Rothman, Decision theoretic approaches to sensor management, DTIC # AB B172227, Wright-Patterson AFB, OH, Feb. 1993.

[25] C. Giraud and B. Jouvencel, Sensor selection in a fusion process: a fuzzy approach, in: Proceedings of the IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems, Las Vegas, Nevada, 1994, pp. 599-606.

[26] P.G. Gonsalves and G.J. Rinkus, Intelligent fusion and asset management processor, in: Proceedings of the International Conference on Information Technology, Syracuse, NY, 1998, pp. 15-18.

[27] N. Gordon, A. Marrs and D. Salmond, Sequential analysis of nonlinear dynamic systems using particles and mixtures, in: W. Fitzgerald, R. Smith, A. Walden, and P. Young, eds. Nonlinear and nonstationary signal processing, Chapter 2, Cambridge University Press, Cambridge, 2001.

[28] P. Greenway and R. Deaves, Sensor management using the Decentralised Kalman Filter, in: Sensor Fusion VII, Proceedings of the SPIE, Vol. 2355, Boston, MA, 1994, pp. 216-225.

[29] G. Hager and H.F. Durrant-Whyte, Information and multi-sensor coordination, in: J.F. Lemmer and L.N. Kanal, eds. Uncertainty in Artificial Intelligence 2, Elsevier Science Publishers B.V., North-Holland, 1998, pp. 381-394.

[30] K.J. Hintz, GMUGLE: A goal lattice constructor, in: Proceedings of the SPIE International Symposium on Aerospace/Defense Sensing & Control, Vol. 4380, Orlando, FL, 2001, pp. 324-327.

[31] K.J. Hintz and G.A. McIntyre, Information instantiation in sensor management, in: Proceedings of the SPIE International Symposium on Aerospace/Defense Sensing & Control, Vol. 3374, Orlando, FL, 1998, pp. 38-47.

[32] K.J. Hintz and G.A. McIntyre, Goal lattices for sensor management, in: Proceedings of the SPIE International Symposium on Aerospace/Defense Sensing & Control, Vol. 3720, Orlando, FL, 1999, pp. 249-255.

[33] K.J. Hintz and E.S. McVey, Multi-process constrained estimation, IEEE Transactions on Systems, Man, and Cybernetics 21 (1) (1991) 434-442.

[34] K. Johansson and P. Svensson, Submarine tracking by means of passive sonobuoys, Technical Report of Swedish Defence Research Establishment, FOA-R—97-00440-505—SE, 1997.

[35] K. Jöred and P. Svensson, Target tracking in archipelagic ASW: a not-so-impossible proposition?, in: Proceedings of Underwater Defence Technology Europe 98, London, 1998, pp. 172-175.

[36] S. Julier and J.K. Uhlmann, General decentralized data fusion with covariance intersection (CI), in: D. Hall and J. Llinas, eds. Handbook of Data Fusion, Chapter 12, CRC Press, 2001, pp. 12-1--25.

[37] M. Kalandros and L.Y. Pao, Controlling target estimate covariance in centralized multisensor systems, in: Proceedings of American Control Conference, Philadelphia, PA, 1998, pp. 2749-2753.

[38] M. Kalandros and L.Y. Pao, Randomization and super-heuristics in choosing sensor sets for target tracking applications, in: Proceedings of the IEEE Conference on Decision and Control, Phoenix, AZ, 1999, pp. 1803-1808.

[39] K. Kastella, Discrimination gain to optimize detection and classification, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 27 (1) (1997) 112-116.

[40] W. Komorniczak and J. Pietrasinski, Selected problems of MFR resources management, in: Proceedings of the International Conference on Information Fusion, 2000, pp. WeC1-3--8.

[41] S. Kristensen, Sensor planning with Bayesian decision analysis, Ph.D. Dissertation, Laboratory of Image Analysis, Aalborg University, Aalborg, Denmark, 1996.

[42] T.W.E. Lau and Y.C. Ho, Super-heuristics and their applications to combinatorial problems, Asian Journal of Control 1 (1) (1999) 1-13.

[43] J. Lindner, R.R. Murphy and E. Nitz, Learning the expected utility of sensors and algorithms, in: Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Las Vegas, Nevada, 1994, pp. 583-590.

[44] W.S. Lovejoy, A survey of algorithmic methods for partially observable Markov decision processes, Annals of Operations Research 28 (1991) 47-66.

[45] R.C. Luo, A.M.D. Shr and C.-Y. Hu, Multiagent based multisensor resource management, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems, Victoria, B.C., Canada, 1998, pp. 1034-1039.

[46] R. Malhotra, Temporal considerations in sensor management, in: Proceedings of the IEEE National Aerospace and Electronics Conference, 1995, pp. 86-93.

[47] R. Malhotra, E. Blasch and D. Johnson, Learning sensor-detection policies, in: Proceedings of the IEEE National Aerospace and Electronics Conference, Wright Laboratory, Dayton, OH., 1997, pp. 769-776.

[48] G.A. McIntyre, A comprehensive approach to sensor management and scheduling, Doctoral Dissertation, George Mason University, Fairfax, FA, 1998.

[49] G.A. McIntyre and K.J. Hintz, An information theoretic approach to sensor scheduling, in: Proceedings of the SPIE International Symposium on Aerospace/Defense Sensing & Control, Vol. 2755, Orlando, FL, 1996, pp. 304-312.

[50] 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, Vol. 3068, Orlando, FL, 1997, pp. 250-260.

[51] G.A. McIntyre and K.J. Hintz, Sensor measurement scheduling: an enhanced dynamic, preemptive algorithm, Optical Engineering 37 (2) (1998) 517-523.

[52] J.M. Molina Lopez, F.J. Jimenez Rodriguez and J.R. Casar Corredera, Fuzzy reasoning for multisensor management, in: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 1995, pp. 1398-1403.

[53] J.M. Molina Lopez, F.J. Jimenez Rodriguez and J.R. Casar Corredera, Symbolic processing for coordinated task management in multi-radar surveillance networks, in: Proceedings of the International Conference on Information Fusion, 1998, pp. 725-732.

[54] S. Musick and R. Malhotra, Chasing the elusive sensor manager, in: Proceedings of the IEEE National Aerospace and Electronics Conference, 1994, pp. 606-613.

[55] G.W. Ng, K.H. Ng and L.T. Wong, Sensor management – control and cue, In: Proceedings of the International Conference on Information Fusion, 2000, pp. WeB1-16 --21.

[56] G.W. Ng and K.H. Ng, Sensor management --- what, why and how, Information Fusion 1 (2) (2000) 67-75.

[57] D. Penny, The automatic management of multi-sensor systems, in: Proceedings of the International Conference on Information Fusion, 1998, pp. 748-755.

[58] D. Penny and M. Williams, A sequential approach to multi-sensor resource management using particle filters, in: Proceedings of the SPIE International Conference on Signal and Data Processing of Small Targets, 2000, pp. 598-609.

[59] C.L. Pittard, D.E. Brown and D.E. Sappington, SPA: Sensor placement analyzer, An approach to the sensor placement problem, Technical Report, Department of Systems Engineering, University of Virginia, 1994.

[60] C.G. Schaefer and K.J. Hintz, Sensor management in a sensor rich environment, in: Proceedings of the SPIE International Symposium on Aerospace/Defense Sensing & Control, Vol. 4052, Orlando, FL, 2000, pp. 48-57.

[61] W. Schmaedeke, Information based sensor management, in: Signal Processing, Sensor Fusion, and Target Recognition II, Proceedings of the SPIE, Vol. 1955, Orlando, FL, 1993, pp. 156-164.

[62] W. Schmaedeke and K. Kastella, Information based sensor management and IMMKF, in: Proceedings of the SPIE International Conference on Signal and Data Processing of Small Targets, Orlando, FL, 1998, pp. 390-401.

[63] D. Simon, Minimax Filtering, http://www.innovatia.com/software/papers/minimax.htm, 2000.

[64] A.F.M. Smith and A.E. Gelfand, Bayesian statistics without tears: a sample-resampling perspective, The American Statistician 46 (2) (1992) 84-88.

[65] J.F. Smith III and R.D. Rhyne II, A fuzzy logic algorithm for optimal allocation of distributed resources, in: Proceedings of the International Conference on Information Fusion, 1999, pp. 402-409.

[66] J.F. Smith III and R.D. Rhyne II, A fuzzy logic resource manager and underlying data mining techniques, in: Proceedings of the International Conference on Information Fusion, 2000, pp. WeB1-3 --9.

[67] A. Steinberg, Problem-solving approach to data fusion, in: Proceedings of the International Conference on Information Fusion, 2001, pp. FrB2-17--26.

[68] A. Steinberg and C. Bowman, Revisions to the JDL data fusion model, in: D. Hall and J. Llinas, eds. Handbook of Data Fusion, Chapter 2, CRC Press, 2001, pp. 2-1--19.

[69] D. Strömberg and G. Pettersson, Parse tree evaluation - a tool for sensor management, in: Proceedings of the International Conference on Information Fusion, 1998, pp. 741-747.

[70] P. Svensson, Information fusion in the future Swedish RMA defence, Swedish Journal of Military Technology 69 (3) (2000) 2-8.

[71] P. Svensson, K. Johansson and K. Jöred, Submarine tracking using multi-sensor data fusion and reactive planning for the positioning of passive sonobuoys, in: Proceedings of Hydroakustik 1997, Stockholm. Technical Report of Swedish Defence Research Establishment, FOA-R--97-00552-409--SE.

[72] K.A. Tarabanis, P.K. Allen and R.Y. Tsai, A survey of sensor planning in computer vision, IEEE Transactions on Robotics and Automation 11 (1) (1995) 86-104.

[73] R. Washburn, A. Chao, D. Castanon, D. Bertsekas and R. Malhotra, Stochastic dynamic programming for far-sighted sensor management, IRIS National Symposium on Sensor and Data Fusion, MIT Lincoln Laboratory, Lexington, 1997.

[74] R. Wesson et al, Network structures for distributed situation assessment, in: A. Bond and L. Gasser, eds. Readings in Distributed Artificial Intelligence, Morgan Kaufmann, 1988, pp. 71-89.

[75] C.C. White, A survey of solution techniques for the partially observed Markov decision processes, Annals of Operations Research 32 (1991) 215-230.

[76] F.E. White, Managing data fusion system in joint and coalition warfare, in: Proceedings of the European Conference on Data Fusion, Malvern UK, 1998, pp. 49-52.

[77] M. Williams, P. Svensson and K. Jöred, Multi-sensor tracking by particle-filter-based predictive sensor deployment, Manuscript, 2001.

[78] Z. Zhang and K.J. Hintz, OGUPSA sensor scheduling architecture and algorithm, in: Proceedings of the SPIE International Symposium on Aerospace/Defense Sensing & Control, Vol. 2755, Orlando, FL, 1996, pp. 296-303.