

CHASING THE ELUSIVE SENSOR MANAGER

Stan Musick and Raj Malhotra
Wright Laboratory, Avionics Directorate

ABSTRACT

In a modern aircraft, data fusion is the process by which data about the environment are gathered, combined, reasoned over, and presented to the pilot. Determining which data to gather is obviously important to achieving effective data fusion. But the need for data depends on uncertain, interrelated and dynamic factors. This fact has pushed data-gathering determination beyond the ability of the human and led researchers to study structured decision-aiding systems called sensor managers. This paper discusses sensor management, focusing first on the problem it poses in a modern tactical aircraft and then on attributes that would be desirable in an effective sensor manager. Several techniques that offer promise are discussed.

INTRODUCTION

Modern tactical aircraft carry multiple sensors, each sensor having controls that provide options for its employment. Although most sensors are potentially synergistic, it is a challenging task to choose control options that will make the sensors cooperate in a coherent manner to achieve mission objectives.

To illustrate, suppose a tactical aircraft is over enemy airspace and is carrying an active radar and a passive Electronic Surveillance Measure (ESM) system. Let us say that this aircraft's unified track picture shows that two of the five objects being tracked are closing rapidly, could pose a serious threat, but are unidentified (see Figure 1). The ESM system provides better

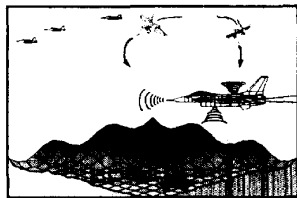


Figure 1 Example

ID than the radar, while the radar provides superior spatial fixes. Both the ID and kinematics of the two objects must be refined before either object can be engaged. However, the two sensors cannot be used simultaneously because radar emissions interfere with the reception spectrum of the ESM. To complicate matters further, there is a pressing need to search a nearby sector where the mission pre-brief showed lethal ground-based threats.

Even though the illustration is a simple one, it is already apparent that resolving conflicting demands on the sensors presents serious dilemmas for the sensor manager (SMgr). The main dilemmas derive from the questions of *what* information is most worth knowing and *how* it should be obtained. When one considers how the complexity of such questions magnifies as more sensors, objects and constraints enter the picture, it becomes clear that this could be a very large and troubling problem.

The goal of sensor management (SMgmt) is to integrate sensor usage to accomplish specific mission objectives at high performance levels. Adaptive control of the sensors (their data gathering and data processing tasks) can afford earlier detection, improved tracking, and more reliable identification, and can do these things despite natural or manmade variations in the environment.

This paper has five sections: **introduction**, this section; **background**, describes where we are today and how we got here, and highlights critical problems; **structure and relationships**, defines the structure of the SMgr and places the software within an integrated architecture; **functions, principles, problems**, describes the environment in some detail, offering impressions about the types of solutions that will be effective; **techniques**, describes some promising approaches and identifies where they might fit in a solution.

BACKGROUND

Aircraft have carried sensors to probe the environment surrounding them for decades, so why has SMgmt just now become important? How were sensors managed in the past and has something changed to make these old ways obsolete? Is the battle situation today any different from yesterday, and if so, what are the implications for sensor use?

Forty years ago, the sensors carried by military aircraft were discrete black boxes, often added piecemeal to existing aircraft as new capabilities emerged. Sensors of that era, which were fewer in number than today, were managed exclusively by the pilot (or some aircrew operator). The pilot alone decided which

sensor to use when, he moded, pointed and controlled each one, and he interpreted its display. The signal environment was less crowded then and the variety of threats was smaller. The pilot's out-the-window observations, along with his interpretation of the sensor displays, provided all the data needed to make decisions. He relied heavily on himself and, perhaps rightly, considered his sensors to be secondary.

The sensors and displays of 40 years ago were often inaccurate, unreliable, and overlapped in function. These shortcomings were recognized early by sensor developers, and a long evolutionary process ensued that produced the integrated avionics systems of today. Modern systems are extremely able and diverse, providing performance characteristics heretofore only dreamed of: e.g. all weather, jam resistant, large surveillance volume, emission controlled, improved in accuracy, aperture agile, and multi-function displays. Even though a performance goal may be achievable by a particular modern sensor, more often the greater unsolved problems involve recognizing how to choose between alternative goals and how to integrate their achievement with conflicting demands from other sensor assets. "The ability to control the invocation of sensor assets supportive of a particular mission is of key operational importance; however, due to the way sensor technologies have been developed, the management of these important assets are at best fragmented, sensor-specific, and are not integrated across sensors." [Denton et al., 89]

An important by-product of sensor evolution is the increase in the quantity of data that is produced. With the increase in sensors and sensor modes, this is inevitable. A good example is tactical radar which has evolved from its inception in World War II as a precision-bombing/navigation-aid system (two modes) to today's systems having 10 or more modes, some of which can operate "simultaneously" in interleaved fashion. As data quantities increase and control choices multiply, workload increases exponentially and eventually even the most able pilots begin to miss important opportunities or fail to recognize critical situations. Pilots are obviously more comfortable thinking about the tactical objectives that sensors can achieve than about the fine details of sensor operation. At some point, pilots welcome the assistance that decision automation can provide.

Another factor to consider is the demanding character of the missions that tactical aircraft must perform. For example, F-117 stealth aircraft were used to attack multiple threats over Baghdad during the Gulf War. These missions were successful despite that platform's exposure to a dense, lethal and rapidly-

changing threat environment. Complex environments like the Gulf cause contentions for sensor assets that can overwhelm their capacities. It is likely that high-risk, multi-target deployments will endanger both stealthy and non-stealthy platforms again.

In the recent past and continuing today, SMgmt systems for tactical aircraft have been constructed using a variety of ad hoc methods. Most often these systems employ rule-based approaches and rely on the pilot to make many real-time deployment decisions. Typically today, automated SMgmt has been limited to scheduling resources to maintain a sufficient rate for performing tasks, with target priorities fixed before the mission. With this type of management, sensors and related processing assets are allocated to achieve an update rate sufficient for maintaining continuity on existing tracks and ensuring the desired probability of detection on new tracks in search sectors. Generally, tactical aircraft lack facilities for coordinating needs between sensors except in a few special circumstances (e.g. alignment to a common boresight or cross-sensor cueing). [Steinberg, 93] In short, no attempt is being made today to provide any consistent conceptual framework for supporting adaptive inter-sensor control functions. An important consequence is the difficulty this causes in utilizing system-level requirements to build a SMgr that could derive best performance from the available assets.

We conclude that the 40-year-old model for SMgmt has changed little, even though sensors proliferate and deployment practices grow more challenging. This model still relies heavily on the pilot to make right choices, and assumes that his judgement and training will be able to handle the increasing workload and overcome the growing complexities of the hardware and the situation. We believe this model is approaching its limits. This paper explores some alternatives.

STRUCTURE AND RELATIONSHIPS

Within the last decade, many problems in SMgmt were recognized and have now been addressed in a variety of focused studies and modeling efforts. Although there has been no definitive work to date, it is generally acknowledged that SMgmt is a complex problem where solutions will probably be evolutionary, not revolutionary. This section looks at this research as it relates to SMgr architectures, and summarizes the important results.

The SMgr is just one element of a "data fusion" system which itself is but one component of a com-

plete avionics system. First we address the SMgr's internal structure as a multi-level system decomposed along hierarchal lines; we see *how* a SMgr could be constructed. Then we present the SMgr as one element of a widely-recognized data fusion model; we see *where* the SMgr lies in the bigger picture of tactical sensing.

Internal Composition: This portrayal is a composite based on several contributions to SMgr design. [Waltz & Llinas, 90; Denton et al., 93; Popoli, 92; Stanley, 92; Bier and Rothman, 88] Most researchers organize the logical functions of the SMgr into a multi-level system. We have seen two-, three- and four-level structures, with levels usually arranged along hierarchal lines. The highest level sets policy, the intermediate level(s) plans and decides, while the lowest level establishes the detailed sensor behavior that will execute the decisions of the higher levels. Such hierarchal arrangements have advantages: they distribute the computational load across multiple processors; they separate the job along natural lines, grouping parts with similar rates and minimizing communications; and they provide a practical design that aids integration and test by maintaining smaller, less-complex parts.

By far, the major research challenges in SMgmt involve finding effective means for organizing and carrying out the upper-level planning and decision functions. These functions are commonly divided into three logical phases that occur sequentially in a coordinated but decoupled arrangement. These three functions are option generation, prioritization, and scheduling. Figure 2 diagrams a typical SMgr system, showing how these functions relate and where several

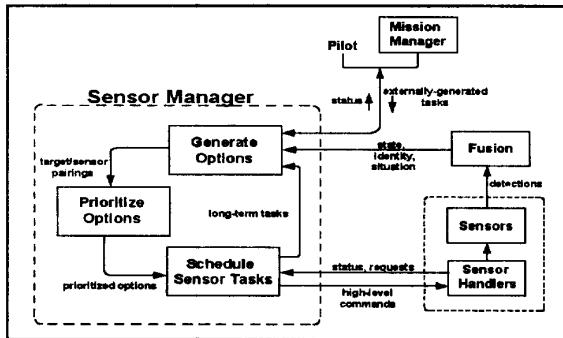


Figure 2 Sensor manager task flow

major accessory functions arise. We briefly discuss this figure. See [Kober et al., 92].

Option generation involves pairing a task with a particular sensor asset (that could perform that task)

to form a candidate for sensor action, e.g. use sensor 3 to search sector B. This task involves determining what information is needed, what sensor actions will capture it, and which sensors are available to perform these actions. Prioritization involves arranging the list of option pairs in a preferred order. Generally, some optimization technique will be used for this step. Such techniques introduce a metric, construct an associated cost function, and apply an (optimal) algorithm that selects the options that minimize the cost function. Finally, scheduling transforms sensor asset requests into a detailed sensor sequencing plan that allocates sensor operations to time slots. Scheduling considers deadlines (hard or soft), durations, synchronicity, etc. The SMgr communicates its requests to a sensor handler, that part of each sensor that regulates waveform generation, tuning, aperture control, deconfliction, dwell strategy, and similar matters.

External Composition: The complete avionics system supports all of the real-time needs of the aircraft. Current work in PAVE PACE, the fourth generation avionics architecture concept, shows these major functional partitions: mission, vehicle, pilot/vehicle, stores, com/nav, maintenance/support, and tactical sensing (includes data fusion). This section presents the data fusion model and the place of the SMgr therein.

Data fusion is the process by which multi-source data is combined to provide the clearest picture possible of the current situation. Data sources to be combined include onboard and offboard sensors and databases,

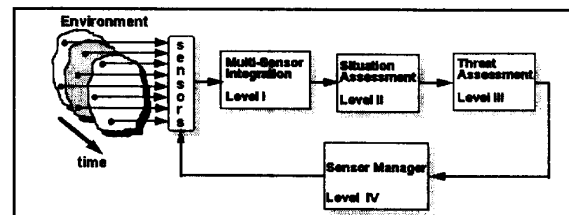


Figure 3 Data fusion model

and pilot observations. The fusion process includes operations such as detection, registration, association, estimation, inference, and prediction. The products of data fusion include kinematic state and identity estimates at the lower levels, and assessments of the overall tactical situation along with event prediction at the upper levels.

The Data Fusion Group of the Joint Directors of Laboratories (see [Waltz & Llinas, 1990]) has developed a general model of the data fusion process to facilitate discussion of data fusion problems and issues. We adopt that model for this discussion. The

four levels of the data fusion process are shown in Figure 3. Thick straight arrows, indicating primary information flow, connect data processing functions at various levels. The curved arcs suggest potential communication paths beyond those in the original model (authors' embellishments). At the lowest level are the sensors which measure kinematic and attribute data for objects in the environment. Level 1 of the data fusion process, *Multisensor Integration*, aligns the measurements (both spatially and temporally), associates the measurements with the proper objects, and estimates the state and identity of objects based on the measured data. *Situation Assessment* (Level 2) attempts to form an estimate of the current tactical situation through some form of reasoning. *Threat Assessment* (Level 3) infers threat levels and predicts future events based on the picture produced at Level 2 and on knowledge of enemy tactics. *Sensor Management* (Level 4) uses the information from Levels 1-3 to formulate a plan for future sensor use.

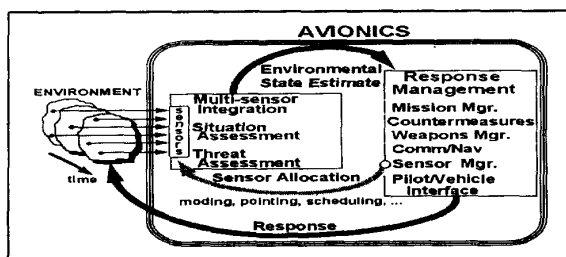


Figure 4 SMgr environment

Figure 4 depicts the relationship of the environment, the data fusion process (shown collapsed into one block), and ownship actions that may affect the environment. Since sensor use may reveal ownship presence and cause the "environment" to react, sensor emissions are shown with actions. And since most of the items in actions are responses to the unfolding situation, they are often called response actions.

FUNCTIONS, PRINCIPLES, PROBLEMS

Summarizing, we can say "The goal of the SMgr is to select the right sensor to perform the right service on the right object at the right time". The ability to accomplish these subgoals supports every **mission function**, including this representative list: fire control, cross-sensor cueing, area search, track, raid assessment, identification-classification-recognition, battle damage assessment, weapon delivery, radar warning, missile warning and tracking, electronic and signal intelligence, and electronic countermeasures.

One should not believe that because the goal statement is simple, the problem is too. SMgmt is a hard problem because it is: a) complex - the number of decisions, random variables, and outcomes is enormous; b) uncertain - inexact ownship kinematics, uncertain kinematics, identity and intent of objects, and noisy measurements; c) contains conflicting objectives - e.g. the need to remain covert conflicts with the need to learn all one can about the situation. In an effort to add breadth to this problem statement, this section lists functions, principles and problems that an effective SMgr system should address.

Primary Functions The primary functions of the SMgmt system are:

- Generate options (sensor/task pairings) for action
- Prioritize options
- Formulate sensor schedule to execute desired actions
- Communicate desired actions to sensor handlers
- Monitor sensor health and performance, respond to sensor feedback, account for sensor availability

These functions require I/O between the SMgr and several partitions in the avionics suite, including: tactical sensing, especially the data fusion system and the sensor handlers; the mission manager which sets priorities as the mission unfolds; the pilot/vehicle interface, to receive and respond to pilot commands; the vehicle, its health and status monitors (see Figure 2).

Principles To effectively accomplish the above functions, these general principles govern SMgr design:

- Plan to use all sensors (offensive & defensive)
- Value long-term goals of survival and success, not just accuracy and identity
- Dynamic environment dictates adaptive methods
- Choose a modeling technique that is mathematically sound, comprehensive, and tractable
- Account for dissimilarities in sensor ability
- Eliminate redundant sensor allocations and take advantage of sensor synergies
- Provide for emission controls (passive and low probability of intercept modes)
- Achieve iteration rates in planning that keep pace with all environment changes
- Shed load gracefully when sensor burden hits limits
- Consider adaptive-length planning horizons

In the fourth bullet, "comprehensive" means able to represent all relevant knowledge, such as the uncertainties associated with both tracking and identifying.

The second bullet forces the SMgr to seek a balance between the best estimate (no emission restrictions) and a good result later (may not emit), a balance that may only show up if the technique is non-myopic (see last bullet).

Problems As it performs the above functions, the SMgr will be constrained by these real-time problems:

- Random sensor failures
- Finite sensing and processing resources
- Enemy interference and spoofing actions
- Enemy evasion tactics

For an interesting account of practical problems and other "innovation barriers" to SMgmt infusion, we recommend [Denton et al., 93].

TECHNIQUES

The objectives of today's SMgmt research are to identify the requirements for, to develop, and to evaluate a technology base of methods and algorithms (we call the pair a technique) for deploying and controlling sensor assets in all mission phases. A SMgmt technique must assign sensors to selected tasks, designating mode, look direction, duration and required information, while accounting for all available information about the situation. SMgmt techniques are key to establishing a valid picture of the current situation, a picture that both pilot and mission manager can use.

At this time, no one technique dominates or even has a large following. Although much remains to be done, there are several techniques that we think show promise. This section discusses four of these, placing each one in the architecture that was discussed above, mentioning their good and bad points, and listing open questions about them.

Information Theoretic [Hintz, 91; Llinas et al., 92; Schmaedeke, 93] all address SMgmt from an information theoretic perspective. Schmaedeke builds on the work of Hintz and Llinas to produce a general technique for optimizing the pairing of sensors to tracking tasks using an information-based metric. An important attribute of this interesting technique is that it does not require the specification of preferences, and therefore avoids the arbitrariness of decision-theoretic formulations.

Schmaedeke presents results for a simple simulation of the technique which show an improvement in tracking accuracy from using the optimal policy. The

technique chooses optimal sensor/target pairings for each scan by using the Kalman filter evolution equations to predict the entropy reduction of all feasible sensor/target pairings, subject to constraints on the capacity of each sensor. The information gain for each pairing is computed as the ratio of the entropy without an update to the entropy given an update is made with the sensor of that pairing. A linear programming routine is used to find the global optimum subject to sensor tracking capacities.

Although Schmaedeke addressed only sensor assignments for tracking, the technique is flexible and already shows promise of being expanded to handle the search and identification tasks as well. Of course, there are open questions about this technique including: its sensitivity to violations of modeling assumptions (e.g. non-Gaussian errors); its potential to suggest a subset of likely pairings to reduce computations (at present, all feasible pairings are evaluated); its computational burden; and its ability to balance conflicting mission goals such as the need to acquire targets but remain covert. Information theoretic approaches do not directly support "scheduling" as defined in Figure 2.

Artificial Neural Networks Artificial Neural Networks (ANNs) represent a unique, biologically-inspired strategy for solving complex problems. The present day plethora of work in this area results primarily from ANN abilities to mimic human intelligence. Accordingly, these systems work well in many areas where traditional computers fail: ANNs are particularly promising in *artificial perception* tasks such as speech recognition and machine vision. They have been used in applications involving function approximation, optimization, prediction, and automatic control. In some cases they have performed impressively and in others, disappointingly. This mixture of failures and successes leaves many questions about their utility in sensor management.

The basic component of a neural network is the neuron model. Figure 5 shows the general structure of a simplified neuron model. The neuron receives a set of input signals, x_i , which are multiplied by a corresponding set of weighting factors, w_i . The weighted input signals are summed and presented as the argument to a nonlinear *activation function*, F . The magnitude of the weights effect the sensitivity of the output to particular inputs. The activa-

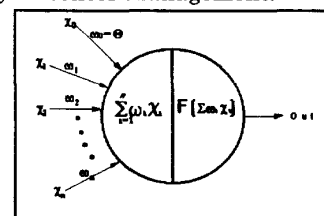


Figure 5 Neuron model

tion function produces the output as a nonlinear mapping of the input vector.

The real power of a neural network emanates from its architecture (fine grain parallelism) and its ability to learn associations. It is fundamentally a *vector mapper*, which maps input vectors into output vectors through some nonlinear transformation. This mapping is parameterized by the network weights which are adjusted in the learning process. The vast majority of network models are variations on two principal topologies; *feed-forward* and *feed-back (recursive)* networks.

A feedforward ANN is shown in Figure 6. This network has an input layer, two hidden layers, and an output layer. The number of neurons in each layer and the number of hidden layers are architectural design decisions that are often application dependent. The feedforward network structure produces a static mapping between the input vector, μ , and the output vector, X as described in Figure 6.

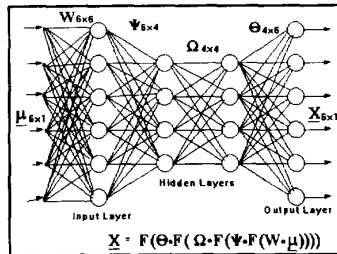


Figure 6 Feedforward net

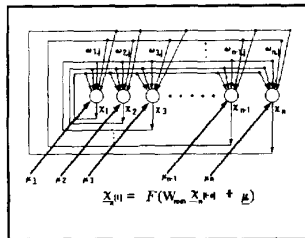


Figure 7 Recursive net

In the feedback network of Figure 7, each neuron receives a weighted output from all the neurons in the single-layer net. This system receives an input vector u and iterates according to the equation $x_i = F(\sum w_{ij} x_j + u_i)$. The

output of a recursive ANN is described by a trajectory of vectors over time, rather than by a single vector in the feedforward case.

ANNs may prove to be useful in several parts of the data fusion process. Their potential in signal estimation and event prediction make them a candidate for multi-sensor integration, and situation/threat assessment. But for SMgmt, their promise stems from their ability to learn an optimization heuristic over some input space. The true intercoupled, stochastic nature of the problem may, in some cases, render optimization using conventional techniques extremely difficult. Neural networks have been shown to accommodate nonlinearities, as well as non-gaussian and non-sta-

tionary processes. Also, the possibility of on-line learning holds great promise for SMgmt applications.

Decision Theoretic Decision theory has enjoyed renewed interest over the last 15 years in applications where difficult decisions must be made under uncertainty. One reason for this interest is the emergence of influence diagrams, a graphical modeling technology that supports decision analysis methods. It is sometimes said that the influence diagram is the tool that operationalizes the theory. There have been successful decision analysis applications in diverse fields including aerospace, engineering, business, management, and medicine. Furthermore, with quality support software becoming more available, these powerful techniques are expected to find greater use in problems of all sizes and complexities.

An influence diagram is a graph that contains nodes connected by directed arcs. Nodes represent decisions, uncertainties, and preferences while directed arcs convey relationship or influence. For example, if two uncertain nodes are joined by a directed arc, the specified relationship is that of conditional dependence. Probability distributions are used to quantify uncertainty and utility functions quantify preferences. Given an influence diagram with its probability distributions and utility functions completely specified, the reasoning task is to solve for the optimal decision policy, that policy which maximizes expected utility. Inference algorithms are able to carry out this reasoning task, thereby specifying how each decision should be made.

SMgmt can be viewed as a decision making process that operates with uncertain data for the purpose of gathering information. As such, SMgmt fits perfectly in the domain of decision analysis where this type of problem is well recognized as a "value of information" problem. An appropriate influence diagram for this problem type is shown in Figure 8 where, by conven-

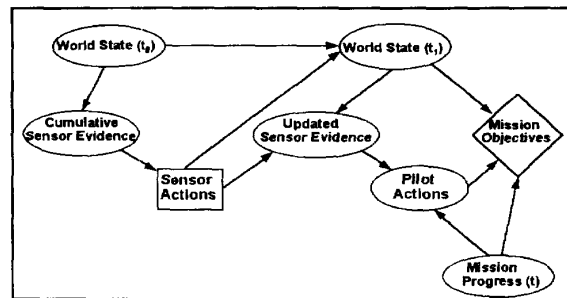


Figure 8 Sensor management influence diagram

tion, squares are decisions, ellipses are uncertainties, and diamonds are preferences. [Fung et al., 1993] The SMgr's reasoning task is to determine which information-gathering actions will maximize the utility.

The "world state" in Figure 8 is that collection of actual (but unknown) conditions that determines what the cumulative evidence will indicate, what the sensors will see, and what utilities will exist as the mission unfolds. The world state has two temporally distinct images to provide for its temporal evolution and for the fact that sensing can alter this state (e.g. active radar warns enemy). The "cumulative evidence" is the estimated probability distribution for the world state before a sensor action is taken. The SMgr has at its disposal a set of possible "actions" (update, identify, ...) for reducing the uncertainty about the world state. The "new evidence" that will be received depends on the action the SMgr chooses and on the world state. This evidence informs but cannot control the "pilot". The uncertain "pilot" node acts as a simple model for coordinating sensor actions with anticipated pilot actions. The "utility" node will have some immediate values (say best information gain or greatest covertness) and some ultimate values (say ownship safe or mission successful) depending on what stage the "mission" is in.

The calculations for a diagram like Figure 8 are quite easy for simple situations. However, the computational burden can go up rapidly if the variables at each node are continuous or if discrete variables are finely hashed. There are also questions about the impact of imperfect probability distributions and arbitrary preference assignments on final outcomes. These questions are certainly valid and sensitivity analysis techniques are available to help answer them. As was true above, decision theoretic methods do not directly support scheduling.

Mathematical Programming Today many important decisions are made by choosing a measure of effectiveness and optimizing it. Deciding how to build an effective SMgr (using a mathematical programming approach) ideally involves three steps. First, one should know, accurately and quantitatively, how the system variables interact. Second, one needs a single measure of effectiveness expressible in terms of system variables. Finally, one should choose those values of the system variables that yield optimum effectiveness. Several techniques have arisen out of management science and optimal control theory which attempt to establish algorithmic methods for solving optimization problems. Together, these are known as Mathematical Programming.

To apply mathematical programming techniques, the problem is expressed as an *objective function*, representing the quantity to be maximized or minimized, and a list of constraints on the system variables. The canonical form is shown below.

$$\begin{aligned} &\text{Minimize/Maximize } z = f(x_1, x_2, \dots, x_n) \\ &\text{such that} \\ &g_i(x_1, x_2, \dots, x_n) = b_i \quad i = 1, \dots, m \end{aligned}$$

The constraints form a *feasible set*, Γ , of possible solutions ($\mathbf{x} \in \Gamma$). The goal is to find the global maximum or minimum of the objective function over the feasible set, Γ . This task is complicated by issues such as linearity of the objective constraining functions, deterministic vs. stochastic relationships, uncertainty representations, nature of probabilistic events, and preference structures. We show linear and nonlinear objective functions in Figure 9 and Figure 10.

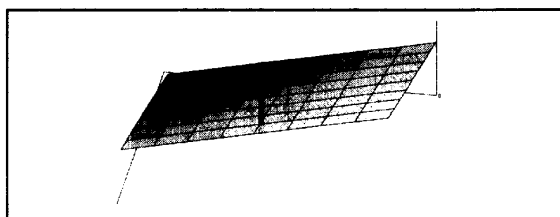


Figure 9 Linear objective function, f_1

Assume that we are to minimize the objective function over some two-dimensional space, $\mathbf{x}=[x_1, x_2]$, and that x_1 and x_2 are completely *controllable*. We may find it much easier to find the global minima of $f_1(x_1, x_2)$ than of $f_2(x_1, x_2)$. Note that we do not have a concise mathematical formulation of these objective functions. In

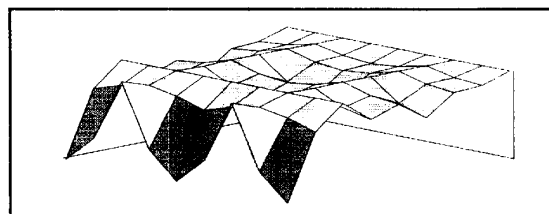


Figure 10 Nonlinear objective function, f_2

the simpler case, we could employ *Linear Programming* techniques such as the simplex method which searches all of the extreme points of the feasible set for a minima. For the nonlinear objective function, we would use *Nonlinear Programming* techniques based upon gradient-descent or some other method which uses local objective function behavior to determine which direction to proceed. There are no guarantees of finding the global minimum. However, in some cases we may be satisfied with a local minimum.

Providing that we can find a deterministic objective function for SMgmt, we may need linear or non-linear programming techniques. But in general, SMgmt will not be favored with linear deterministic objective functions or with system variables that are all controllable, and will also find that many intermediate decisions must be made as the dynamic situation evolves. Imagine Figure 10 above as a static "snapshot" of a dynamic, nonlinear, nonstationary process. The question becomes, "what trajectory in x_1 and x_2 minimizes the objective?". The field of *Dynamic Programming* addresses such questions.

In *Dynamic Programming* we form the problem as follows: **Given** x_0 and the discrete-time dynamic system

$$x_{k+1} = f_k(x_k, u_k, w_k) \quad k = 0, 1, \dots, N-1$$

where x 's represent overall system states, u 's represent system inputs, and w 's are random disturbances (accounting for the stochastic nature of the problem), **Find** some admissible control law, $\pi = \{u_0, u_1, \dots, u_{N-1}\}$ that minimizes the total cost function

$$J_{\pi}(x_0) = E_{w_k} \{g_N(x_N) + \sum_{k=0}^{N-1} g_k[x_k, u_k(x_k), w_k]\}$$

where g 's are scalars denoting the cost at each time step. Note that the total cost is the expected value of the sum of the intermediate costs over the random process w_k . The outcome is dependent upon the disturbance as well as intermediate system states.

The dynamic programming technique decomposes the problem into a sequence of simpler minimization problems that are carried out over the control space, $\pi = [u_1, u_2, \dots, u_n]$, rather than over a space of functions over the current state, x . Although dynamic programming does not begin to resolve the modeling complexities innate to SMgmt, it does establish a mathematically tractable structure to find an optimum sequence of decisions over either a finite or infinite horizon.

REFERENCES

- Beightler, CS, Phillips, DT, Wilde, DJ (1978) *Foundations of Optimization*, Prentice-Hall, Englewood Cliffs, New Jersey.
- Bier, SG and Rothman, P (1988) "Intelligent sensor management for beyond visual range air-to-air

combat", *Proceedings of the NAECON*, May 1988, Dayton, Ohio, 264-270.

- Denton, RV, Alcaraz, EI, Knopow, JP, Llinas, J and Hintz, KJ (1993) "Towards modern sensor management systems", *Proceedings of Sixth Joint Service Data Fusion Symposium*, 14-18 June, Johns Hopkins Univ, MD, 659-678.

- Fung, R, Horvitz, E and Rothman, P (1993) "Decision theoretic approaches to sensor management", DTIC # AD B172227, Feb 1993, WPAFB, Ohio.

- Hintz, KJ (1991) "A measure of the information gain attributable to cueing", *IEEE Transactions on Systems, Man, and Cybernetics*, 21(2):434-442.

- Kober, W, Meyer, D and Thomas, J (1992) "An analytic-based sensor management software system", Wright Laboratory report WL-TR-93-1018, Dec 1992, WPAFB, Ohio.

- Llinas, J, Hintz, KJ and Beale, GO (1992) "Applications of automatic control theory to sensor scheduling", Naval Warfare Center, Aircraft Division, NAWCADWAR-93002-50, 30 Sep 1992, Warminster, Pennsylvania.

- Popoli, R (1992) "The sensor management imperative" in: Y Bar-Shalom (ed), *Multitarget-Multisensor Tracking: Applications and Advances*, Vol II, Artech House, Boston, Massachusetts.

- Schmaedeke, Wayne (1993) "Information based sensor management", *SPIE Conference on Target Recognition and Data Fusion II*, May 1993, Orlando, Florida.

- Stanley, J (1992) "PAVE PACE, Concepts for sensor integration", *National Aerospace and Electronics Conference*, 18-22 May 1992, Dayton, Ohio.

- Steinberg, A (1993) "Sensor and data fusion" in: Robinson (ed), *Emerging Systems and Technologies*, Vol 8 of *The Infrared and Electro-Optical Systems Handbook*, Acceta and Shumacher (eds), SPIE Optical Engineering Press, Bellingham, Washington, .

- Waltz, E and Llinas, J (1990) *Multisensor Data Fusion*, Artech House, Boston, Massachusetts.

- Wasserman, PD (1993) *Advanced Methods in Neural Computing*, Van Nostrand Reinhold, New York.