SONAR TRACKING

Over the past three decades, a large number of investigators have contributed to the theoretical and practical aspects of *sonar tracking*. Our intent in this article is to expose key developments that give the reader a sufficiently complete overview of many topics in tracking with particular emphasis on sonar tracking. Comprehensive treatment of these topics can be found in Blackman (1), Waltz and Llinas (2), Antony (3), Bar-Shalom (4,5), and Bar-Shalom and Fortmann (6). The invention of the *Kalman filter* is perhaps the single, most influential technological advance that has made possible the current mature state of sonar tracking. Necessarily, our exposition includes a discussion of the Kalman filter. Although during the early stages of development the Kalman filter provided a computationally revolutionary mechanism for estimating the state of a tracked object, the practical real-life applications in sonar tracking were limited to single-target tracking because of the limitations imposed by the computing capabilities of the processing hardware. In sonar tracking, the source of information consisted of only passive acoustic sensors. As computing resources became more readily available, multiple-target tracking capabilities were developed. Thus emerged the concept of developing an overall integrated surveillance scene containing multiple targets. Capabilities were developed for processing information from a variety of sensor systems, in addition to acoustic sensors, to develop a sonar scene.

Multisensor, multitarget tracking systems have been routinely used in a variety of applications during the past two decades. In many applications, and especially in sonar environments, because of their high clutter character, it became evident that a single hypothesis regarding the scene used to represent the interpretation of all the inputs from all the sensor systems was not adequate. In Ref. 7, a new approach was proposed to represent the information using multiple simultaneous interpretations in the form of multiple-scene hypotheses. Thus began a new era in sonar tracking, with a number of approaches developed to deal with ambiguity, efficiency, and accuracy. Our discussion includes a fairly complete review of many issues related to the *multihypothesis tracking* (MHT) subject.

Much of the early development of algorithms and techniques in sonar tracking focused on the topic of tracking the state of individual objects. In the parlance of the more encompassing domain of *data fusion,* the individual target tracking is considered to be occurring at Level 1—also known as *object refinement*—of information processing. In the more recent past, the focus of these developments has shifted to the higher level of information content. The concepts of *situation refinement, threat refinement,* and *process refinement* were the natural evolutionary steps in the development of tracking. The related theoretical topics include use of both knowledge-based techniques and fuzzy-neural representations, and new developments in sensor management and fusion strategies. Much of the discussion that follows presents these areas in more detail.

CORRELATION, ASSOCIATION, AND FUSION

The central problem in multisensor, multitarget sonar tracking is the data association problem of partitioning contacts into tracks and false reports. This problem is formulated as multiscan processing, it is valid for either centralized fusion or decentralized tracking. The mathematical formulation of the data association problems is separated from the algorithms that solve this problem. Before discussing problem formulation, a brief review of data association follows.

General approaches to single-scan processing include nearest neighbor, global nearest neighbor (solved by the two-dimensional assignment problem), probabilistic data association (PDA), and joint PDA (JPDA). The former two approaches are real-time, but decisions once made are irrevocable, leading to poor track estimation, to fragmentation, and even to loss of tracks. The latter two approaches have been successful for tracking in heavy clutter, but have had difficulties with closely spaced targets. Another class of methods is called deferred logic, or multiscan, processing. The most popular method is called multiple-hypothesis tracking (MHT). These methods are well-suited to tracking a potentially large number of targets in a cluttered environment.

The fundamental problem for multiscan processing is to maximize the probability of data partition into tracks and false reports (8–10). The data association problems for multisensor and multitarget tracking are generally posed as maximizing the posterior probability of the set of tracks (given the data) according to

$$\text{Maximize}\{P(\pi = \Pi|Z^N)|\Pi \in \Pi^*\} \tag{1}$$

where Z^N represents N data sets or scans, π is a partition of indices of the data (and thus induces a partition of the data into tracks), Π^* is the finite collection of all such partitions, Π is a discrete random element defined on Π^*, $P(\pi = \Pi|Z^N)$ is the posterior probability of a partition π being true given the data Z^N, and P is the probability measure of a partition π of the cumulative data Z^N into tracks and false reports. For the assignment formulation, under independence assumptions, this problem is equivalent to finding a solution of

$$\text{Minimize} - \ln\left[\frac{P(\pi|Z^N)}{P(\pi^0|Z^N)}\right] \equiv \sum_{i_1=0}^{M_1}\cdots\sum_{i_N=0}^{M_N} c^N_{i_1\ldots i_N} z^N_{i_1\ldots i_N}$$

where $c^N_{i_1 \ldots i_N}$ is the negative log of the likelihood ratio $L^N_{i_1 \ldots i_N}$,

$$z_{i_1\ldots i_N} = \begin{cases} 1 & \text{if } (z_{i_1}, \ldots, z_{i_N}) \text{ are assigned to the track} \\ 0 & \text{otherwise} \end{cases}$$

is a zero-one variable, and π^0 is a reference partition consisting of N false reports. The constraints for this problem impose the requirement that each report z_{i_k} from scan k must be assigned to exactly one track of data $(z_{i_1}, \ldots, z_{i_N})$.

This problem is precisely what all approaches to association and fusion try to solve. The difficulty is that the problem

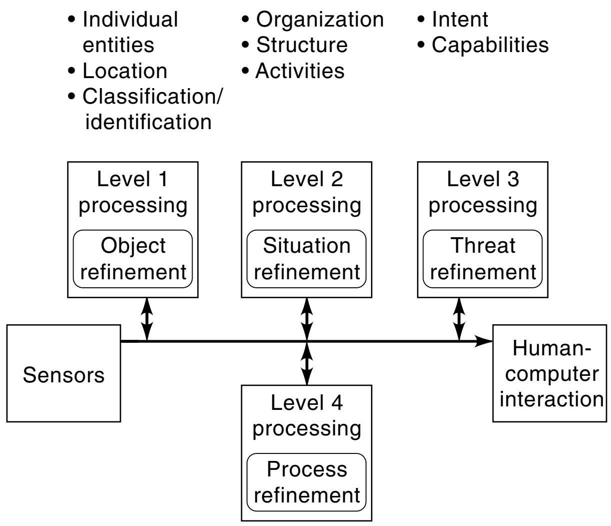

Figure 1. High-level JDL four-level data fusion functional model.

is nonpolynomial (NP)-hard, so that any algorithm that solves it is NP-hard, and all known algorithms that solve the problem optimally require a time that grows exponentially with the size of the problem. A fundamental problem with sequential processing is that data association decisions are irrevocable. MHT corrects this problem by allowing changes in the data association over the last N scans.

Now consider N data sets $Z(k)$, $k = 1, \ldots, N$, with M_k reports $\{z^k_{i_k}\}^{M_k}_{i_k=1}$, respectively, and let Z^N denote the cumulative data set defined by $Z(k) = \{z^k_{i_k}\}^{M_k}_{i_k=1}$ and $Z^N = \{Z(1), \ldots, Z(N)\}$, respectively. The data sets $Z(k)$ may represent different objects, and each data set can be generated from different sensors. For track initiation, measurements are partitioned into tracks and false alarms. In track maintenance, which uses a moving window over time, one data set will be tracks and remaining data sets will be scans of measurements. In sensor-level tracking, the objects to be fused are tracks from multiple sensors. In centralized fusion, the objects may be a combination of measurements that represent targets or false reports and tracks that have already been filtered; the problem is to determine which measurements emanate from a common platform.

Fusion Strategies

The Joint Director of Laboratories (JDL) model divides the data fusion processing into four levels. All four levels of processing use and share the same data and information, as shown in Fig. 1. Processing in Level 1 deals with object refinement, which is positional, kinematic, and attribute fusion of single tracks within the ocean. In Level 2 situation refinement processing, a description or interpretation of the current relationships among objects and events in the context of the environment is developed. Threat assessment, in Level 3 processing, develops a threat-oriented perspective of the data to estimate enemy capabilities, identify threat opportunities, estimate enemy intent, and determine levels of danger. Finally, Level 4 processes refinement processing monitors and evaluates the ongoing fusion process to refine the process itself, for example, by tasking sensors to gather additional information or resolve ambiguities.

The JDL model defines the process of expanding from traditional statistical/mathematical techniques of fusion to include artificial intelligence for data assimilation, correlation, and abstraction, resulting in a "hybrid" system that uses cognitive processing technologies to add intelligence to the process of data fusion and determination of target identification. The advanced fusion technology analyzes the situation, as a human operator would, with awareness of the situation beyond the data that is being reported by the current sensors. With this awareness, the system can make inferences based on knowledge of the environment, the current state of the situation, threat tendencies, and the assets it has available to help resolve target identification.

Kalman Filtering

At the heart of data fusion algorithms is a tracking algorithm, typically a Kalman filter. Under certain conditions (11,12), the Kalman filter provides an optimal estimator that minimizes the mean square error. In addition, the Kalman filter can be implemented in an efficient recursive manner. In the case where a nonlinear relationship exists between the measurement vector and the state vector, for example, a range/bearing measurement where the tracking coordinates are x-y, an extended Kalman filter (EKF) or an iterated EKF (IEKF) provides a suboptimal approximation.

A summary of the Kalman filter is presented in Table 1. The models describe the motion of the target, including the uncertainty of the model represented by system noise ω_k and the relationship between the state and the measurement. When a new measurement is processed, the first step of the Kalman filter is to predict the latest state estimate and its

Table 1. Summary of the Nonlinear Iterated Extended Kalman Filter

Models

$$\boldsymbol{x}_{k+1} = \Phi_{k+1}\boldsymbol{x}_k + \omega_{k+1}$$

$$\boldsymbol{z}_{k+1} = h(\boldsymbol{x}_{k+1}) + \epsilon_{k+1}$$

where ω_k is $N(0, \boldsymbol{Q}_k)$ and ϵ_k is $N(0, \boldsymbol{R}_k)$

Prediction

$$\boldsymbol{x}_{k+1|k} = \Phi_{k+1}\boldsymbol{x}_{k|k} \tag{1}$$

$$\boldsymbol{P}_{k+1|k} = \Phi_{k+1}\boldsymbol{P}_{k|k}\Phi^T_{k+1} + \boldsymbol{Q}_{k+1} \tag{2}$$

Iterative Updates—For $i = 0, 1, 2, 3, \ldots$

Note: The extended Kalman filter is obtained by setting $i = 0$.

$$\hat{\boldsymbol{z}}_{k+1,i} = h_{k+1}(\boldsymbol{x}_{k+1,i}) \tag{3}$$

$$\boldsymbol{H}_{k+1,i} = \left.\frac{\partial h_{k+1}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=x_{k+1,i}} \tag{4}$$

$$\boldsymbol{r}_{k+1,i} = \boldsymbol{z}_{k+1} - \hat{\boldsymbol{z}}_{k+1,i} - \boldsymbol{H}_{k+1,i}(\boldsymbol{x}_{k+1|k} - \boldsymbol{x}_{k+1,i}) \tag{5}$$

$$\boldsymbol{C}_{k+1,i} = (\boldsymbol{H}_{k+1,i}\boldsymbol{P}_{k+1|k}\boldsymbol{H}^T_{k+1,i} + \boldsymbol{R}_{k+1})^{-1} \tag{6}$$

$$\boldsymbol{K}_{k+1,i} = \boldsymbol{P}_{k+1|k}\boldsymbol{H}^T_{k+1,i}\boldsymbol{C}_{k+1,i} \tag{7}$$

$$\boldsymbol{x}_{k+1,i+1} = \boldsymbol{x}_{k+1|k} + \boldsymbol{K}_{k+1,i}\boldsymbol{r}_{k+1,i} \tag{8}$$

$$\boldsymbol{P}_{k+1,i+1} \doteq (\boldsymbol{I} - \boldsymbol{K}_{k+1,i}\boldsymbol{H}_{k+1,i})\boldsymbol{P}_{k+1|k} \tag{9}$$

Initial Conditions

$$\boldsymbol{x}_{k+1,0} = \boldsymbol{k}_{k+1|k}$$

covariance (or uncertainty) of the time of the measurement [Eqs. (1) and (2) of Table 1]. The next step is to estimate the expected measurement by using the predicted measurement, as specified by Eq. (3). Next, the residual between the estimated and actual measurement is computed [Eqs. (4) and (5)], along with its estimated covariance [Eq. (6)]. Finally, the Kalman filter gain [Eq. (7)] is computed and used to update the state estimate and its covariance [Eqs. (8) and (9)].

Bearing-Only Tracking. One of the fundamental problems of sonar tracking is performing localization from a set of measurements obtained from a passive sensor, i.e., given a set of passive bearings or line-of-bearing measurements, develop an estimate of the target position and velocity. Much effort has been focused on the issues of observability (the inherent information contained in the measurement set to provide a localization) and coordinate systems (13,14). The fundamental result on observability states that the relative motion between the observing platform must be nonlinear. In simplest terms, if the target is on a constant course/constant speed leg, the observer must maneuver at least once before a localization solution can be computed. Details on localization can be found in Refs. 15 and 16.

In order to isolate the problem of observability, researchers have investigated the impact of coordinate systems used in the Kalman filter. One popular coordinate system is the inverse polar coordinate system (β, $\dot{\beta}$, r, $\dot{r}/r$), where β and $\dot{\beta}$ are the bearing and bearing rate, respectively, and r and $\dot{r}/r$ are the range and normalized range rate, respectively (17).

Gaussian Sum. Based on the fact that most density functions can be approximated arbitrarily close by a sum of Gaussian density functions, the Gaussian sum approach (18,19) provides an attractive alternative to an inverse polar coordinate system for bearing only tracking. Given a density function f with a finite number of discontinuities, f can be approximated arbitrarily close by a finite sum of Gaussian density functions. Let

$$f_k(\boldsymbol{x}) = \sum_{k=1}^{K} \alpha_k N(\boldsymbol{x} - \mu_k; \Sigma_k)$$

where N is the Gaussian density function with mean μ_k and covariance Σ_k, with

$$\sum_{k=1}^{K} \alpha_k = 1$$

and $\alpha_k \geq 0$ for all k. Then, selecting α_k, μ_k, Σ_k, and K, f_K can approximate f to an arbitrary degree of closeness. Given a line of bearing, a sum of Gaussians can be used to approximate a bearing wedge, as depicted in Fig. 2. In addition, environmental information, such as direct path and convergence zone propagation, can be directly modeled as a Gaussian sum. Thus, the sum of Gaussians can be used to model the nonlinear bearing measurement. An advantage of the Gaussian sum approach is that a linear Kalman filter can be used by running K filters, one for each term in the Gaussian sum.

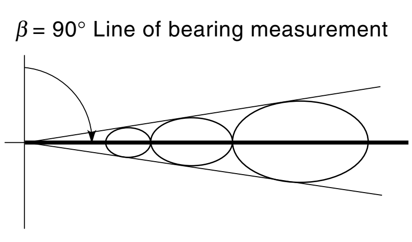

Figure 2. Gaussian sum approximation to a line of bearing measurement.

Table 2. Assignment Problem Example

		New			Old	
		c_1	c_2	c_3	c_4	c_5
New system track	T_1	0.5	0.0	0.0	—	—
New system track	T_2	0.0	0.15	0.0	—	—
New system track	T_3	0.0	0.0	0.03	—	—
System track	T_4	0.4	0.0	0.0	—	—
System track	T_5	0.3	0.2	0.0	—	—
System track	T_6	0.05	0.8	0.7	—	—
System track	T_7	0.0	0.7	0.6	—	—
System track	T_8	—	—	—	0.3	—
System track	T_9	—	—	—	—	0.8

Optimal Assignment Strategies

We assume that part of the overall sonar system is a preprocessor that associates measurements. For example, an automatic line tracker on a gram provides the association of a specific narrowband signal source. Thus, contacts and measurements can be of two types: first, the sensor system provides an association of some of the measurements into contacts, in which case the reported contacts in a scan are either new (not previously reported) or old; second, the sensor system does not perform association, in which case all the contacts reported in a scan are new.

The measurement data-to-track assignment problem is depicted in Table 2. The assignment matrix is structured such that the first M_2 rows consist of possible new tracks and the last M_1 rows consist of tracks from the current hypothesis. Note that the upper $M_2 \times M_2$ block is simply a diagonal matrix (a measurement can be assigned to only one new track).

The objective of the assignment function is to find a "best" set of solutions. The optimal solutions to assignment problems are given in Refs. 1 and 20. The "solution vector" assigns each data measurement to some track in the hypothesis; each data point either updates an existing track within the hypothesis or is assigned to a new track. In this case, the optimal assignment is c_1 to T_4, c_2 to T_7, c_3 to T_6, c_4 to T_8, and c_5 to T_9. Note that for c_2, the optimal is T_7, not the largest likelihood that is associated with T_6, because the c_2/T_7 and c_3/T_6 assignment pair has a higher likelihood (0.49) than the c_2/T_6 and c_3/T_7 assignment pair (0.48).

Multiple-Hypothesis Tracking

The multiple-hypothesis tracker (MHT) data fusion algorithm with clustering is, in essence, a two-layer algorithm: the first (lower) layer consists of a multiple-hypothesis algorithm that carries alternative hypotheses of how the data is partitioned into tracks, and the second layer consists of cluster management that breaks the problem into noninteracting, disjoint clusters. The lower layer multiple-hypothesis algorithm matches data to tracks, updates tracks, generates hypotheses, and manages algorithm resources (both tracks and hypotheses). The second layer, cluster management, monitors each set of hypotheses to ensure that tracks within a cluster do not interact with tracks that are in other clusters.

A typical MHT implementation is depicted in Fig. 3. The six primary processing functions are:

- Gating (track-data scoring)
- Clustering
- Assignment solution (association, track updating, hypothesis generation)
- N-scan pruning
- Renormalization
- Splitting

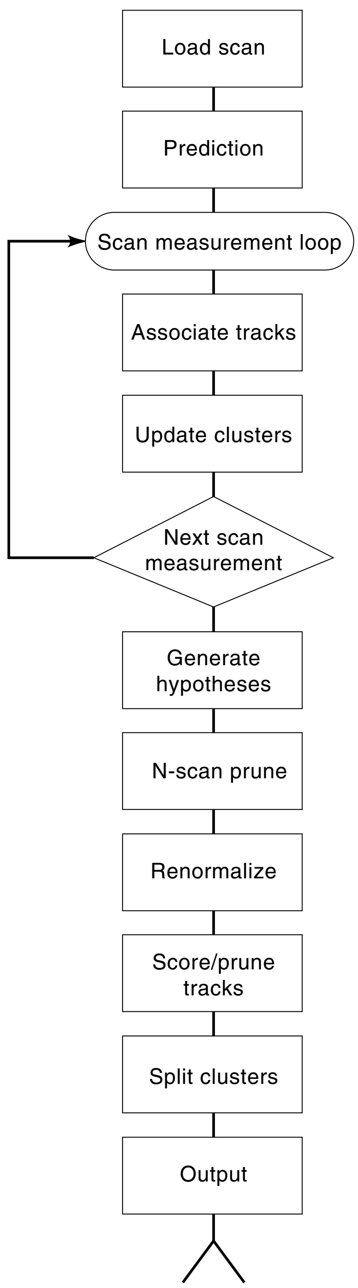

Figure 3. Core algorithm flow chart.

The lower level algorithms of track-data scoring, association, track updating, hypothesis generation, and algorithm resource management have an extended Kalman filter at the heart of the MHT. Track-data scoring is based on the value of the density function of the normalized residual. This score is computed for each existing track and each measurement in a scan, except when the normalized residual itself is larger than a fixed threshold (usually set at four to six standard deviations).

In the association step, a two-dimensional assignment algorithm (such as a modified Munkres algorithm) is used to select the optimal assignment of measurements in a scan to tracks in a hypothesis. Part of this assignment is the determination of a new track for each of the measurements. In order to generate additional hypotheses, the assignment algorithm is run again on the original problem with modified costs of the assignment matrix of the optimal assignment.

Once the optimal and suboptimal solutions are obtained, the hypothesis scores are computed and compared. High-scoring hypotheses are kept for further analysis, whereas low-scoring hypotheses are pruned. A hypothesis score is recursively obtained by multiplying the old hypothesis score by each of the assignment scores of the track-data associations determined by the two-dimensional assignment solutions.

Finally, N-scan pruning is used to accomplish two goals. First, N-scan pruning helps keep the overall number of hypotheses under control. More importantly, N-scan pruning forces a hard decision on all measurements in the $(N - 1)$th oldest scan. Thus, N-scan pruning is a sliding window that allows the MHT algorithm to carry multiple hypotheses on the most current data and make hard decisions on older data (which is based on data up to the current time).

The following assumptions and conditions are made:

1. The measurement data of the scan are valid at the same time t_k.
2. Each measurement comes from a distinct target.

The first assumption is made to simplify the cluster gating implementation; thus, tracks are predicted to the time of the current scan only once for the entire scan of the measurements. This assumption can be relaxed at the cost of additional time for execution. The second assumption is fundamental to hypothesis generation. Because each measurement comes from a distinct target, the number of data association combinations is limited as two measurements from the same scan cannot be put into the same track. Thus, the fundamental number of hypotheses is limited.

Gating. All tracks are predicted to the time of the current scan. The Kalman filter prediction equations are used to extrapolate each track state and error covariance estimate to the time of the current scan using Eqs. (1) and (2) of Table 1. Next, the extrapolated state estimate is used to calculate the predicted measurement vector using the state to measurement transformation Eq. (3) of Table 1.

The normalized residual is computed as

$$r = \boldsymbol{r}^T \boldsymbol{C}^{-1} \boldsymbol{r}$$

where $\boldsymbol{r}$ is the residual vector from Eq. (5) of Table 1, and $\boldsymbol{C}^{-1}$ is the inverse of the residual vector covariance matrix from Eq. (6) of Table 1. The normalized residual r is a χ^2_m statistic with m degrees of freedom, where m is the dimension of the measurement vector z_k. A probability of geometric association $P_g(\boldsymbol{r})$ is computed for a track-to-measurement candidate if the normalized residual passes the gating criterion

$$\chi^2_m \leq n^2$$

where the value of n, the gate size, is a parameter that can be interpreted to mean an $n - \sigma$ track-to-measurement containment.

For each normalized residual that passes the gating criteria, the probability of geometric association $P_g(\boldsymbol{r})$ is computed as the likelihood density function of an $N(\mathbf{0}, \boldsymbol{C})$ normal random variable. This probability is evaluated as

$$P_g(\boldsymbol{r}) = \frac{1}{(2\pi)^{m/2}|\boldsymbol{C}|} \exp(-r/2)$$

where $\boldsymbol{C} = \boldsymbol{HPH}^t + \boldsymbol{R}$ (the residual vector covariance of the Kalman filter equations, see Table 1), m is the measurement dimension and r is the computed normalized residual. Scoring of the new track probability of association is based on the likelihood density function.

Clustering. The basic purpose of clustering is to divide the large data fusion problem into a number of smaller ones that can be solved independently. Each cluster maintains a noninteracting set of tracks and data. Clustering is an adaptive process, driven by the interactions and ambiguity of the incoming data.

Clusters are initiated in two distinct ways. A new cluster is initiated each time a data point is received that does not fall within the correlation gates of any track contained in an existing cluster. The new cluster then contains one hypothesis consisting of a single track (the new track) with a probability of one. In addition, a new cluster is initiated when a given track is contained in all hypotheses of a previous cluster.

In order that clusters remain distinct, they must be combined when a new data point is received that fits with tracks from more than one cluster. Thus, when a data point falls within the correlation gates of two or more clusters, the clusters are merged. New hypotheses are formed from all combinations of the hypotheses in the clusters being merged. The set of tracks and data points in the new "super cluster" is the union of those in the prior clusters. The number of hypotheses in the new super cluster is the product of the number of hypotheses in the prior clusters and the associated probabilities are the products of the prior probabilities.

An explicit example of cluster merging is now presented. Let cluster C_1 contain two hypotheses $H_1^{(1)}$ and $H_2^{(1)}$ with hypothesis scores $p_1^{(1)}$ and $p_2^{(1)}$, respectively. Let cluster C_2 contain three hypotheses $H_1^{(2)}$, $H_2^{(2)}$, and $H_3^{(2)}$ with scores $p_i^{(2)}$, $i = 1, 2, 3$. Then the new merged cluster contains a total of six hypothesis, namely, $H_1^{(1)} \oplus H_1^{(2)}$, $H_1^{(1)} \oplus H_2^{(2)}$, $H_1^{(1)} \oplus H_3^{(2)}$, $H_2^{(1)} \oplus H_1^{(2)}$, etc., where the hypothesis $H_i^{(1)} \oplus H_j^{(2)}$ is formed simply by taking the union of the track sets contained in $H_i^{(1)}$ and $H_j^{(1)}$. The probability of the new hypothesis p_{ij} is computed as the product of the probability of the corresponding hypotheses $p_{ij} = p_i^{(1)} p_j^{(2)}$. Pruning, if necessary, is based on the p_{ij}, and a normalization of the new cluster hypothesis scores is performed.

Assignment Solution. The primary objective of the assignment solution function is to find the "best" set of solutions for each hypothesis in each cluster. The solutions in each cluster are ranked on score, with the lower scoring hypotheses pruned; the top-ranked solutions are then used to generate a set of new hypotheses for the cluster. Any new tracks are initialized and existing tracks are updated.

The score of the assignment is computed as the product of the individual association probabilities. The solution score is computed as the product of the assignment score and the solution's generating hypothesis score. The optimal solution is used to obtain the set of next-best solutions. This step is accomplished by not allowing associations that are in the optimal solution one data point at a time.

The most important aspect of cluster management is the allocation of the number of hypotheses that each cluster is allowed to carry. The solutions in each cluster are ranked on score, with the N_n highest scoring solutions retained; the lower scoring solutions are pruned if the number of solutions is greater than N_n. An adaptive pruning mechanism is also used. Solutions with scores less than an adaptive threshold score are pruned. The adaptive threshold score is computed as a ratio of the top-scoring solution.

New tracks are initialized for each measurement that is not assigned to an existing track in the generating hypothesis. The track states and covariances are initialized according to the type of measurement, e.g., range/bearing, latitude/longitude, or bearing-only. Existing tracks that have a measurement assigned are updated. Eqs. (3)–(10) of Table 1 are used to perform the track update.

***N*-Scan Pruning.** The two primary functions of N-scan pruning are ancestry update and N-scan pruning. Each hypothesis that was generated must have its ancestry updated. Pruning is accomplished by computing the sum of probabilities of current hypotheses that have a common ancestry on the previous N^{th} scan. The ancestor set with the largest probability is kept and all other hypotheses are pruned. Each hypothesis that was generated must have its ancestry updated. Each generated hypothesis H' points to the parent hypothesis H. This ancestry update is kept for the last N scans of measurement data.

Renormalization. Hypothesis scores within a cluster are renormalized such that the sum of the probabilities of all hypotheses within a cluster is one. Track scores are computed for each track by summing the probabilities of the hypothesis in which they occur. After all scoring and pruning is complete, hypothesis scores within each cluster are renormalized such that the probabilities of all hypotheses within a cluster sum to one. This simply involves adding the scores of all hypotheses within a cluster and dividing each hypothesis by the resulting sum. Specifically, let cluster C_i, $i = 1, \ldots, N_C$, contain hypotheses $H_j^{(i)}$, $j = 1, \ldots, N_C^{(i)}$. Let $p_j^{(i)}$ be the probability of hypotheses $H_j^{(i)}$; then the renormalized hypothesis score for hypothesis $H_j^{(i)}$ is

$$\frac{p_j^{(i)}}{\sum_{j=1}^{N_C^{(i)}} p_j^{(i)}}$$

Track Score and Prune. After the hypothesis scores of a cluster are renormalized, a score is computed for each track in the cluster by summing the probabilities of the hypotheses in which the track is contained. Thus, the track score ranges from zero to one; it is equal to one if the track appears in all hypotheses. If a track appears in cluster C_i, then the score $P(T)$ of track T is computed as

$$p(T) = \sum_{T \in H_j^{(i)} \in C_i} p_j^{(i)}$$

where $H_j^{(i)}$ varies over all the hypotheses in cluster C_i, and $p_j^{(i)}$ is the renormalized hypothesis score.

The final step of the MHT algorithm is cluster splitting, which is the process of subdividing an existing cluster into smaller, independent clusters. Clusters are split for two distinct reasons. A cluster is split when a track is contained in all hypotheses of a previous cluster. This track is removed from all hypotheses of the previous cluster and inserted into a single hypothesis in the new cluster. In addition, clusters containing one hypothesis with more than one track are split.

N-Dimensional Assignment

An alternative to MHT, which processes a single scan at a time, is the N-dimensional (ND) assignment approach, which simultaneously solves the assignment problem over N scans of data. For notational convenience in representing tracks, we add a zero index to each of the index sets and a dummy report z_0^k to each of the data sets $Z(k)$, and define a "track of data" as $(z_{i_1}^1, \ldots, z_{i_N}^N)$ where i_k and $z_{i_k}^k$ can now assume the values of 0 and z_0^k, respectively. A partition of the data refers to a collection of tracks of data wherein each report occurs exactly once in one of the tracks of data and such that all data are used; the occurrence of a dummy report is unrestricted. The dummy report z_0^k serves several purposes in the representation of missing data, false reports, initiating of tracks, and terminating of tracks (9,21,24).

Next, under appropriate independence assumptions, the track scores are computed as

$$\frac{P(\pi = \Pi | Z^N)}{P(\pi = \Pi_0 | Z^N)} = L_\gamma = \prod_{i_1 \ldots i_N \in \Gamma} L_{i_1 \ldots i_N}$$

where $L_{i_1 \ldots i_N}$ is the likelihood ratio containing probabilities for detection, maneuvers, and termination as well as probability density functions for measurement errors, track initiation, and termination. Then, with $c_{i_1 \ldots i_N} = -\ln L_{i_1 \ldots i_N}$,

$$-\ln\left[\frac{P(\Pi | Z^N)}{P(\Pi_0 | Z^N)}\right] = \prod_{i_1 \ldots i_N \in \gamma} c_{i_1 \ldots i_N}$$

Expressions for the likelihood ratios $L_{i_1 \ldots i_N}$ can be found in Refs. 7–10, 21. In track initiation, the N data sets all represent reports from N sensors, possibly all the same. For track maintenance, we use a sliding window of N data sets and one data set containing established tracks. The formulation is the same as in the preceding except that the dimension of the assignment problem is now $N + 1$.

With the zero-one variable $z_{i_1 \ldots i_N}$ if $i_1 \ldots i_N \in \pi$ and 0 otherwise, the problem can be formulated as the following N-dimensional assignment problem:

Minimize

$$\sum_{i_1=0}^{M_1} \cdots \sum_{i_N=0}^{M_N} c_{i_1 \ldots i_N} z_{i_1 \ldots i_N}$$

Subject to

$$\begin{aligned}
&\sum_{i_2=0}^{M_2} \cdots \sum_{i_N=0}^{M_N} z_{i_1 \ldots i_N} = 1, \; i_1 = 1, \cdots M_1 \\
&\sum_{i_1=0}^{M_1} \cdots \sum_{i_{k-1}=0}^{M_{k-1}} \sum_{i_{k+1}=0}^{M_{k+1}} \cdots \sum_{i_N=0}^{M_N} z_{i_1 \ldots i_{k-1} i_{k+1} \ldots i_N} = 1 \\
&\quad \text{for } i_{k+1} = 1, \ldots, M_{k+1}, \text{ and } k = 1, \ldots, N-1 \\
&\sum_{i_1=0}^{M_1} \cdots \sum_{i_{N-1}=0}^{M_{N-1}} z_{i_1 \ldots i_N} = 1, i_N = 1, \cdots, M_N \\
&z_{i_1 \ldots i_N} \in \{0, 1\} \quad \text{for all } i_1, \ldots, i_N
\end{aligned} \tag{2}$$

Efficient algorithms for solving Eq. (2) are specified in Refs. 22–24.

Probabilistic Data Association

A popular method for tracking in highly cluttered environments is joint probability data association (1,4). At time k, let the measurements z_i^k, $i = 1, \ldots, m_k$ fall within the association gate of a track and let

$$P_g(\boldsymbol{r}_i) = \frac{\beta^{m_k}(1 - P_\mathrm{D})}{(2\pi)^{m/2}|\boldsymbol{C}_i|} \exp(-r_i/2)$$

where $\beta = P_\mathrm{NT} + P_\mathrm{FA}$ is the sum of the new track and false alarm probabilities and P_D is the probability of detection, $\boldsymbol{C}_i = \boldsymbol{HPH}^t + \boldsymbol{R}_i$ (the residual vector covariance of the Kalman filter equations, see Eq. (4) of Table 1), m is the measurement dimension, and r_i is computed normalized residual. We assume that the new track and false alarm rates follow a Poisson distribution. For convenience, let $\boldsymbol{z}_0$ represent a missed measurement and $P_g(r_0) = \beta^{m_k}(1 - P_\mathrm{D})$ the likelihood that none of the measurements inside of the gate were generated by the track.

Let

$$\beta_i(k) = \frac{e^{(-r_i/2)}}{b + \sum_{j=1}^{m_k} e^{(-r_i/2)}}$$

$$\beta_0(k) = \frac{b}{b + \sum_{j=1}^{m_k} e^{(-r_i/2)}}$$

where

$$b = \beta(2\pi)^{m/2}(1 - P_\mathrm{D})|\boldsymbol{C}|^{1/2}$$

and $\boldsymbol{C} = \boldsymbol{C}_i$ is assumed to be constant for all measurements within the gate. Then the updated mean is

$$\boldsymbol{x}_{k|k} = \sum_{i=0}^{m_k} \beta_i(k) \boldsymbol{x}_{k|k}(\boldsymbol{z}_k)$$

and the covariance is

$$\mathbf{P}_{k|k} = \beta_0(k)\boldsymbol{P}_{k|k-1} + [1-\beta_0(k)][\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k]\boldsymbol{P}_{k|k-1} + \tilde{\boldsymbol{P}}_k$$

where

$$\tilde{\boldsymbol{P}}_k = \boldsymbol{K}_k\left[\sum_{i=1}^{m_k}\beta_i(k)\boldsymbol{r}_i(k)\boldsymbol{r}_i^T(k) - \boldsymbol{r}(k)\boldsymbol{r}^T(k)\right]\boldsymbol{K}_k^T$$

is the "correction" term to the standard Kalman filter and

$$\boldsymbol{r}(k) = \sum_{i=1}^{m_k}\beta_i(k)\boldsymbol{r}_i(k)$$

is the weighted residual.

When additional information is available, such as the amplitude information from a passive narrowband source, improved performance can be achieved. A probabilistic data association-based maximum likelihood estimator, using amplitude information, has been developed (24a). Although a small improvement in the Cramer–Rao lower bound is achieved, Monte Carlo simulations showed gains in increased accuracy and a reduction in false tracks, especially at low signal-to-noise ratios.

CLASSIFICATION AND IDENTIFICATION

Beyond the localization of tracks, the classification of the individual contacts is an important aspect of the overall sonar tracking problem.

Bayesian Inference Networks

Integrating or fusing attribute information over time is an important processing mechanism required to derive target identity. A taxonomic hierarchy is the perfect mechanism to maintain belief over time for every identity level. For simplicity, a taxonomic hierarchy, one form of Pearl tree or Bayesian evidential reasoning algorithm, is presented. A complete discussion of taxonomic hierarchies and general Bayesian networks is presented in Ref. 25.

Pearl Tree Structure. A Pearl tree is an N-node, as opposed to binary, tree structure. Each tree node represents a specific hypothesis. Each hypothesis can be divided into subhypotheses, or be a subhypothesis itself. Every node is initially assigned an *a priori* measure of belief reflecting the prior probability that the hypothesis is true. These measures of belief range from 0.0, reflecting no confidence, to 1.0, reflecting complete confidence. The measure of belief of the tree's root node sum is always 1.0. In general, the probability of a specific node equals the sum of the probabilities of its subnodes.

Figure 4 illustrates a simple Pearl tree for target identification. In this example, the number inside the node name represents the node probability. Clearly, the evidence suggests that the target is most likely a Hostile Submarine target, and of all possible Hostile Submarine platform types, most likely a nuclear submarine. Even in this small example, every identity level is enumerated.

In the example illustrated in the preceding, many different paths exist through the tree that represent an independent and mutually exclusive set of hypotheses. These sets of hypotheses are called "cuts." Some examples of a cut are the sets {Subsurface, Surface} and {Hostile Submarine, Neutral Submarine, Friendly Submarine, Surface}. A cut is considered valid if every element in the cut is always independent of the others and the probabilities of every element in the cut sum to unity. An example of an invalid cut is the set {Nuclear, Surface}. This cut is invalid because not all nodes are represented.

Pearl Tree Evidence Propagation. Sensor-specific attribute data and geometric heuristic information are used as evidence to determine target identification. The likelihood ratio λ_i measures the degree to which the evidence supports or refutes the hypothesis h_i represented by node i. That is, for a piece of evidence e, the likelihood ratio is given by

$$\lambda_i = \frac{\Pr(e|h_i)}{\Pr(e|\text{not } h_i)}$$

Positive support for h_i is given if $\lambda_i > 1.0$; negative support is given for h_i if $\lambda_i < 1.0$. Generally, the likelihood ratios for an entire cut are arbitrarily assigned rather than explicitly computed.

Let $\mathrm{BEL}(h_i)$ be the measure of belief in the hypothesis represented by node i. Then, for every node i in a cut, an updated belief is obtained by

$$\Pr'(h_i) = \alpha\lambda_i\Pr(h_i) \tag{3}$$

where α is a normalization factor given by

$$\alpha = \left[\sum_i \lambda_i \Pr(h_i)\right]^{-1} \tag{4}$$

Every subnode j below node i in a tree is updated by

$$\Pr'(h_j) = \alpha\lambda_i\Pr(h_j) \tag{5}$$

Each supernode k above the nodes in a cut is updated by summing the updated beliefs of those nodes in a cut that are subnodes of supernode k. That is,

$$\Pr'(h_k) = \sum_{\substack{i \text{ a subnode} \\ \text{of } k}} \Pr'(h_i) \tag{6}$$

As an example, suppose the nuclear hostile subsurface node from the tree shown in Fig. 4 is injected with the likelihood value $\lambda = 4$. The updates to the nodes are shown outside the nodes in the figure.

Fuzzy Rule-Based Fusion Strategies

In cluttered and uncertain sonar environments, the information provided by the sensor systems is not precisely specified. In many situations, one or more components of the sensor information are supplied with nonquantitative qualifiers. Fuzzy representations can be used efficiently in these situations to extract the information for data fusion purposes (26–28). Here, a fuzzy representation is simply the mapping from an input measurement space to an output measurement us-

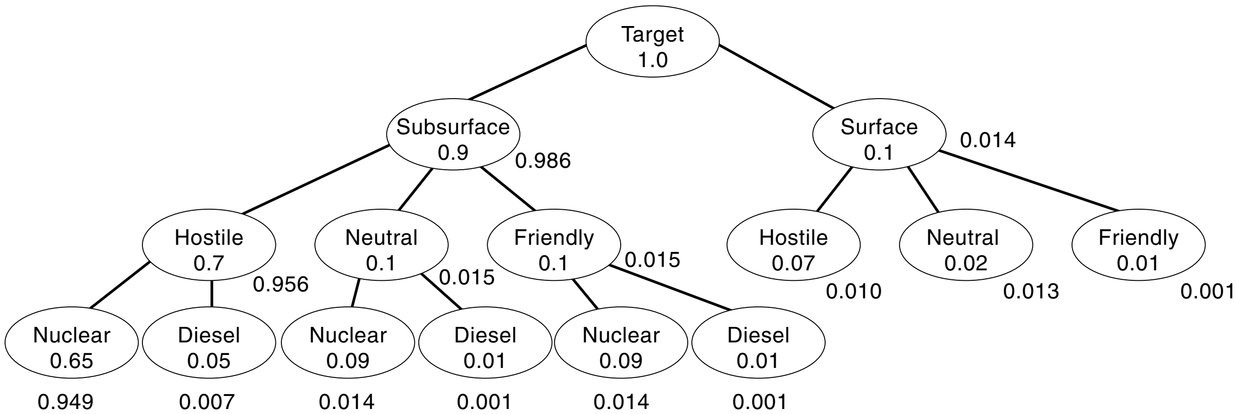

Figure 4. Simple Pearl tree for target identification.

ing linguistic variables. It gives us the ability to model imprecisions by incorporating qualitative components into a quantitative analysis. The use of fuzzy logic in data association or correlation (29–34) is a more recent development in sonar tracking. Some of the relevant techniques for association are summarized in what follows.

The Use of Fuzzy Measures. Fuzzy measures provide a mechanism for assigning belief or plausibility to a set of crisp events. We can structure the data correlation problem to fit within the framework of fuzzy measure theory. Furthermore, this treatment of data as fuzzy sets can be incorporated in sonar tracking problems through the multiple-hypothesis fusion architecture to be described later. Here, fuzzy membership functions and traditional statistical methods are used to represent each crisp event. The primary mechanism is to use a fuzzy implementation of the extended Kalman filter (EKF) discussed earlier. This approach provides a powerful method for data representation through the use of the nonquantitative and unpredictable character of sensor measurements.

In an environment in which clutter exists, to reduce the effects of the clutter measurements without losing the information contained in the true measurements from the target, a weighting scheme for the measurements that uses fuzzy logic has been developed by Priebe and Jones (35). The fuzzy filter defined in Ref. 35 uses only the distance information for the fuzzy membership. However, the technique is derived on the basis of general rules, and not rules specifically related to this distance measure. To incorporate more rules, we can simply incorporate them with existing rules via fuzzy logic.

We define the Mahalanobis distance for each observation as

$$\mu_{k,i} = \boldsymbol{r}_{k,i}^T(\boldsymbol{R}_k + \boldsymbol{HPH}^T)^{-1}\boldsymbol{r}_{k,i}$$

where $\boldsymbol{r}_{k,i}$ is the residual from sensor i. This distance serves as the universe of discourse for the fuzzy predicate. For example, the fuzzy predicate "similar" is defined as

$$f_{\text{similar}}(\mu_{k,i}) = e^{-\mu_{k,i}/2}$$

A "valid" membership function is created to reduce the computational requirements; the membership function is defined as

$$f_{\text{valid}}(\mu_{k,i}) = \begin{cases} 1 & \mu_{k,i} \le \gamma_k \\ 0 & \text{else} \end{cases}$$

Invoking the fuzzy intersection of the two membership functions in the preceding, the resulting membership function is one that is the minimum of the two membership functions

$$f_{\text{similar}\cap\text{valid}}(\mu_{k,i}) = \min[f_{\text{similar}}(\mu_{k,i}), f_{\text{valid}}(\mu_{k,i})]$$

The term $\beta_0 = 1$ is defined as the output if no intersections are valid. The defuzzified output residual that is subsequently fed into the Kalman filter update equations becomes

$$\boldsymbol{r}_k = \frac{\sum_{i=1}^{m_k} f_{\text{valid}\cap\text{similar}}(\mu_{k,i})\boldsymbol{r}_{k,i}}{\beta_0 + \sum_{i=1}^{m_k} f_{\text{valid}\cap\text{similar}}(\mu_{k,i})} = \frac{\sum_{i=1}^{m_k} \beta_{k,i}\boldsymbol{r}_{k,i}}{\sum_{i=1}^{m_k} \beta_{k,i}}$$

where $\beta_{k,i}$ is a weighting function based on the fuzzy intersection of the "similar" and the "valid" membership function.

Processing Fuzzy Measurements. A fuzzy extended Kalman filter (EKF) is an extension to the standard EKF in which a set of fuzzy rules and models are used. We discuss two fuzzy EKF algorithms. The first algorithm incorporates only fuzzy measurements. During the processing of the state estimates, the algorithm defuzzifies the measurement information and computes a crisp state estimate. The second algorithm permits all variables to be fuzzy numbers. The resulting state estimate vector comprises fuzzy numbers.

Using Fuzzy Measurements in the Extended Kalman Filter. A general way to admit a fuzzy set in the place of the measurement vector over a general class of estimation procedures is introduced in what follows. This technique, first proposed by Watkins (36), provides reasonable answers for situations in which the actual measurement is rendered ambiguous. The basic premise of this work is to incorporate a fuzzy membership function and the concept of a fuzzy estimator into the Kalman filter.

Given a new measurement $\boldsymbol{z}$, an estimator maps the measurement data to an estimate. Also, we assume that a suitable fuzzy membership function $m_{\text{adj}}(\boldsymbol{z})$ has been defined *a priori* for the measurement type. The fuzzy set m_{adj} is said to be *informative* if the relation

$$0 < \int m_{\text{adj}}(z)\,dz < \infty \tag{7}$$

holds where the integral is taken over $\Re^n$ for the n-vector $\boldsymbol{z}$.

For an informative membership function, we define the estimator x, using the normalized membership function as a weighting function, as

$$E(x) = \frac{\int x(z) m_{\text{adj}}(z)\,dz}{\int m_{\text{adj}}(z)\,dz} \tag{8}$$

Equation (8), which averages the estimates against the given fuzzy set, becomes our estimate. Although the EKF estimator is commonly used, the estimator x can be any desired estimator. Because it is normalized with respect to the membership function, Eq. (8) is a moment generating function. The following two results provide a basis for the fuzzy estimator in Watkins (8).

Result 1. Given an informative fuzzy set m_{adj} and an estimator x that has a finite first moment with respect to m_{adj}, Eq. (8) estimates the same quantity as does x. Moreover, this estimate is optimal in the sense of average squared-error with respect to m_{adj}.

Result 2. The estimate of Eq. (8) reproduces the original estimator x evaluated at $\boldsymbol{z}$ when the input data is crisp, and when the point $\boldsymbol{z}$ is the "limit" of a sequence of membership functions that converge to atomic measure at $\boldsymbol{z}$.

With Eq. (8) and Results 1 and 2, we can now proceed to develop the results to apply fuzzy measurements to an EKF. The linearity of the Kalman filter with respect to the measurement trivializes the implementation of the EKF to handle fuzzy data as shown in the following results.

Result 3. Let m_{adj} be an informative fuzzy set and $\boldsymbol{x}(\boldsymbol{z})$ be an update algorithm integrable with respect to m_{adj}. Then, if $\boldsymbol{x}(\boldsymbol{z})$ is a matrix-linear function of the vector input $\boldsymbol{z}$, the estimator defined by Eq. (8) is just the given function applied to the first moment vector $\text{mom}_1(m_{\text{adj}})$ of m_{adj}.

In order for the EKF routine described in the preceding paragraphs to be implemented, a set of both antecedent membership functions and consequence membership functions must exist for the sensor measurement. Examples of these membership functions used in sonar and ground tracking are given in Lobbia (37). The resulting implementation for the fuzzy EKF is achieved in the following three steps.

Step 1. Apply the fuzzy inference. By using the knowledge about the premise of the fuzzy measurement and the consequence membership function, we create the new membership function of the fuzzy conclusion.

Step 2. Compute the mean value of the fuzzy conclusion membership function. This step is the first moment discussed in Result 3 here.

Step 3. The moment computed in Step 2 is the crisp value to be applied to the EKF. From this point, apply the standard EKF algorithm.

A Fuzzy Extended Kalman Filter. In a recent paper, Hong and Wang (38) presented a technique that allows fuzziness to propagate throughout the extended Kalman filter. They believe that, because the measurements are fuzzy, the state will be fuzzy, as will the measurement's noise covariance. This fuzziness then propagates throughout the computed estimates of the EKF.

To evaluate the equations involved with the Kalman filter, it is necessary to avoid the problem that occurs after multiple fuzzy arithmetic operations, in which the fuzziness of the data will continue to grow into an unacceptable range. The following implementation is suggested to avoid this problem.

Step 1. Defuzzify the measurement noise covariance $\boldsymbol{R}^*$ and the error covariance $\boldsymbol{P}^*$.

Step 2. Compute the Kalman gain $\boldsymbol{K}$ by

$$\boldsymbol{K}1_k^* = \boldsymbol{P}^*\boldsymbol{H}^T(\boldsymbol{R} + \boldsymbol{H}\boldsymbol{P}^*\boldsymbol{H}^T)^{-1}$$
$$\boldsymbol{K}2_k^* = \boldsymbol{P}\boldsymbol{H}^T(\boldsymbol{R}^* + \boldsymbol{H}\boldsymbol{P}\boldsymbol{H}^T)^{-1}$$

and then take the intersection of $\boldsymbol{K}1^*$ and $\boldsymbol{K}2^*$.

Step 3. Defuzzify the Kalman gain $\boldsymbol{K}^*$, the measurement $\boldsymbol{z}^*$, and the state estimate $\boldsymbol{x}^*$.

Step 4. Update the state estimate by computing

$$\boldsymbol{x}1^* = \boldsymbol{x}^*_{k|k-1} + \boldsymbol{K}(z - h(\boldsymbol{x}^*_{k|k-1}))$$
$$\boldsymbol{x}2^* = \boldsymbol{x}_{k|k-1} + \boldsymbol{K}^*(z - h(\boldsymbol{x}_{k|k-1}))$$
$$\boldsymbol{x}3^* = \boldsymbol{x}_{k|k-1} + \boldsymbol{K}(z^* - h(\boldsymbol{x}_{k|k-1}))$$

and then take the intersection of $\boldsymbol{x}1^*$,,$\boldsymbol{x}2^*$, and $\boldsymbol{x}3^*$.

Step 5. Update the error covariance by computing

$$\boldsymbol{P}1^* = \boldsymbol{P}^*_{k|k-1} - \boldsymbol{K}\boldsymbol{H}\boldsymbol{P}^*_{k|k-1}$$
$$\boldsymbol{P}2^* = \boldsymbol{P}^*_{k|k-1} - \boldsymbol{K}^*\boldsymbol{H}\boldsymbol{P}_{k|k-1}$$

and then take the intersection of $\boldsymbol{P}1^*$ and $\boldsymbol{P}2^*$.

Step 6. Defuzzify the updated error covariance, the process noise covariance $\boldsymbol{Q}^*$, and the updated state estimation $\boldsymbol{x}_{k|k}$.

Step 7. Compute the error covariance prediction by computing

$$\boldsymbol{P}1^* = \Phi\boldsymbol{P}^*_{k|k}\Phi^T + \boldsymbol{Q}$$
$$\boldsymbol{P}2^* = \Phi\boldsymbol{P}_{k|k}\Phi^T + \boldsymbol{Q}^*$$

and then take the intersection of $\boldsymbol{P}1^*$ and $\boldsymbol{P}2^*$.

Step 8. Compute the state estimation prediction as $\boldsymbol{x}^*_{k+1|k} = \phi(\boldsymbol{x}^*_{k|k})$, and return to Step 1.

Neural Network Algorithms

A commonly occurring situation in sonar tracking is that the dynamics of the target change or become unknown. Therefore, the model representations used in the tracking system must be adaptively adjusted. In recent work, Lobbia and Stubberud (39) have developed an adaptive state estimator that is an EKF augmented by an artificial neural network (ANN). This method was developed for use with control systems where the dynamics of the system were not completely known. The known dynamics were used by the EKF as its dynamical model, while the ANN learned the unmmodeled dynamics of the system. Thus, the neural network-based EKF's overall dynamical system approached that of the true plant. This technique can also be applied to learn the maneuver motion model from the sensor measurements. A detailed development of this technique can be found in Refs. 39–41. A summary of this development is presented here. The general discrete-time model that is applied to the EKF tracking algorithm is given in Table 1.

The motion model of the target $\phi_k(\boldsymbol{x}_k)$ is usually not a fully known quantity, especially during a maneuver. Also, it is not known when the target starts to implement a maneuver. For these reasons there is an error function between the true trajectory of the target $\phi_k(\cdot)$ and the mathematical model was developed to approximate that trajectory $\hat{\phi}_k(\cdot)$ given by

$$\boldsymbol{x}_{k+1} = \varphi(\boldsymbol{x}_k) + \nu_k \tag{9}$$

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k) + \eta_k \tag{10}$$

$$\epsilon_k = \phi_k(\boldsymbol{x}_k) - \phi_k(\boldsymbol{x}_k) \tag{11}$$

Obviously, the smaller the error, the better will be the tracks from the EKF.

Using a simple multilayer feedforward ANN $g_k(\boldsymbol{x}_k, \boldsymbol{w}_k)$, where $\boldsymbol{x}_k$ is the track estimate and $\boldsymbol{w}_k$ is the set of weights of the ANN as a function approximator, we can estimate ϵ_k. Unfortunately, the weights are a set of unknown quantities that must be identified. To train the weights for the ANN, we use a variation of the EKF training paradigm of Singhal and Wu (42). We will not reconstruct their results here but, simply stated, we construct an EKF to estimate the states of the dynamical system

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k \tag{12}$$

with the residual

$$\epsilon_k - g_k(\boldsymbol{x}_k, \hat{\boldsymbol{w}}_k) \tag{13}$$

The resulting states of the EKF become the weights of the ANN. By integrating our ANN into the *a priori* mathematical model, we let our total model become the sum of the approximate model $\hat{\phi}_k(\boldsymbol{x}_k)$ and our ANN approximation $g_k(\boldsymbol{x}_k, \boldsymbol{w}_k)$ is given by

$$\boldsymbol{x}_{k+1} = \hat{\phi}_k(\boldsymbol{x}_k) + g_k(\boldsymbol{x}_k, \boldsymbol{w}_k) \tag{14}$$

However, note that Eq. (14) is dependent on the weight estimates of $\boldsymbol{w}_k$. Therefore, we must include the ANN and its training in the EKF algorithm, thus redefining the estimated-state prediction as

$$\overline{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x}_{k+1|k} \\ \boldsymbol{w}_{k+1|k} \end{bmatrix} = \begin{bmatrix} \hat{\phi}_k(\boldsymbol{x}_{k|k}) + g_k(\boldsymbol{x}_{k|k}) \\ \boldsymbol{w}_{k|k} \end{bmatrix} \tag{15}$$

Similarly, we incorporate the ANN into the covariance prediction. We rewrite the error covariance prediction $\boldsymbol{P}_{k+1|k}$ as

$$\boldsymbol{P}_{k+1|k} = \left(\tilde{\Phi} + \frac{\partial g(\boldsymbol{x}_{k|k}, \boldsymbol{w}_{k|k})}{\partial \boldsymbol{x}_{k|k}}\right)_{k|k} \boldsymbol{P} \left(\tilde{\Phi} + \frac{\partial g(\boldsymbol{x}_{k|k}, \boldsymbol{w}_{k|k})}{\partial \boldsymbol{x}_{k|k}}\right)^T + \boldsymbol{Q}_k \tag{16}$$

where $\boldsymbol{x}$ is the augmented state vector of Eq. (15) and

$$\tilde{\Phi} = \begin{bmatrix} \Phi & 0 \\ 0 & \boldsymbol{I}_{\mathrm{w}} \end{bmatrix}$$

where the Jacobian of our *a priori* model Φ is defined by

$$\Phi_{ij} = \left.\frac{\partial \Phi_k(\boldsymbol{x})_i}{\partial \boldsymbol{x}_j}\right|_{\boldsymbol{x}=\boldsymbol{x}_{k|k-1}}$$

The terms of the other EKF equations are augmented to handle the dimensionality increase resulting from the addition of the ANN weights into the state estimation vector. The primary change is the augmentation of the Jacobian $\boldsymbol{H}$ with zeroes, so as not to affect directly the estimated output with the ANN weights.

As Eq. (16) shows, the new EKF is of significantly larger dimension than the standard EKF because of the weight training. This increased complexity can reduce run-time efficiency. However, with efficient programming techniques, we can reduce the computational complexity of the routines. The algorithm also has the advantage of being simply a larger EKF. Thus, we can incorporate the fuzzy capabilities into the algorithm with relatively minimal work. One other problem can exist with this technique. The size of the ANN can affect convergence. If the ANN is too small, it may not have the capability to learn the modeling error. If the ANN is too large, the training can become too slow for useful implementation. A feasible implementation of the neural network-based extended Kalman filter is given in Ref. 39.

SENSOR MANAGEMENT IN FUSION SYSTEMS

Efficient management of sensor resources in a dynamic environment requires optimized coordination of the actions of the controllable sensor system assets available to the platform. In sonar applications, both passive and active assets need to be managed. Passive sensor management involves optimal use of the information content of reports for data fusion and control of the operational and processing environment in which they operate. For active sensors, the management function requires control of the actions of the sensor to focus its attention in desired surveillance space and correct operating conditions. Among the information collection functions of the surveillance process are those that use the sensors to search a desired area for targets, detect and acquire targets, and track acquired targets. In the *search mode,* the sensor systems are given a vague description of the target states; the controlled sensor systems have not detected the target yet, and the sensor controls are generated by the sensor management process using the "null" information to optimize the actions of the sensor system. The "null" information is given to the sensor management process in the form of a report that states that the execution of the control actions dictated by the process resulted in a failure to detect a signal. The *detection mode* is used to transition the control actions from the search mode to the track mode. In the *track mode,* the sensor systems generate positive reports in the form of measurements that are functionally related to the state of the system.

The objectives of the control function are different in the search and track modes. In the search mode, the control process strives to optimize sensor configurations to obtain a first detection. In the track mode, it continually tries to optimize the sensor configuration to avoid a first missed detection. Therefore, ideally the system desires to minimize the time of first detection or to maximize the time for first failure to detect a tracked target. Because of a lack of suitable computational structures for these times, other suitably formulated and tractable measures of performance are used by the control process in obtaining sensor control strategies. In the search mode, the detection probability is one such measure. In the track mode, estimation accuracies are used as a measure of performance.

MATHEMATICAL FORMULATION AND REPRESENTATIONS FOR THE SENSOR MANAGEMENT FUNCTION

The general approach for deriving sensor control strategies consists of the following steps:

1. Optimal processing of information gathered by sensors, to compute statistics that can be used by control algorithms
2. Optimal processing of the preceding statistics to compute search strategies that are used to reconfigure the sensor system operation
3. Reconfiguration of the sensor system using the preceding sensor strategies

In subsequent paragraphs, these steps are described in more detail.

Evolution of Surveillance State and Its Probability Density

The fundamental quantity that underlies this investigation is the *a posteriori* transition density for the state of the target system given all the information up to the current time. A brief discussion of the effects of information on the transition probability density function and rules for its evolution are given in what follows.

Let $x(t)$ denote the state of a target. The dynamics of such a target can be adequately described by a suitable stochastic differential equation

$$dx(t) = \phi(\boldsymbol{x}(t), t)\, dt + g(\boldsymbol{x}(t), t)\, d\beta(t) \tag{17}$$

where $\beta(t)$ is an independent increment process defining the noise. Let I_{t,t_0} denote all the information available at time t. This information is collected by many sensors in the system. Let $y_\alpha(t)$ denote the state of the αth sensor. This sensor generates two types of reports:

1. A *positive* report is given when the sensor detects the target. Over the time interval $[0, t]$ a sensor may detect the target several times; $N_\alpha(t)$ denotes the number of detections reported by the αth sensor over the interval $[0, t]$. Some types of sensors also provide additional information about the target when a detection is made. Let $\boldsymbol{z}_\alpha(t)$ denote the measurement generated by the αth sensor when it detects a target; $\boldsymbol{N}(t)$ and $\boldsymbol{z}(t)$ will be the composite information in the form of vector processes.
2. A *negative* report at time t is one in which no detections are recorded by the αth sensor over the interval $[0, t]$, i.e., $N_\alpha(t) = 0$. Therefore, there is no $\boldsymbol{z}_\alpha(t)$ associated in this case. For the purpose of analysis, the information I_{t,t_0} used in computing the *a posteriori* transition probability density function $p(\boldsymbol{x}(t), t|\boldsymbol{x}(t_0), t_0, I_{t,t_0})$ is equivalently described by the sub-σ-field $G^t_{t_0}$ generated by $\boldsymbol{N}(t)$ and $\boldsymbol{z}(t)$ over the interval $[t_0, t]$. For notational convenience, when $t_0 = 0$ we denote $G^t_{t_0}$ simply by G^t.

The two important questions of concern are:

1. Given all the information I_{t,t_0}, what is the optimal sensor assignment policy, $Y^*_t = \{Y^*(\tau), t \leq \tau \leq t\}$, and
2. Given I_{t,t_0}, and a sensor assignment Y_t, what is the best estimate $\hat{\boldsymbol{x}}(t)$ of the target state $\boldsymbol{x}(t)$?

To consider further the issues stated in the foregoing text requires suitable measures of performance. For the sensor assignment problem, a suitable selection criterion is the resultant detection probability $P_D(Y_t)$. For the state estimation problem, the widely used performance measure is the error covariance associated with the estimate $\hat{\boldsymbol{x}}(t)$. Under certain conditions, maximization of $P_D(Y_t)$ is equivalent to minimization of the error covariance (43).

The first problem to address is a stochastic control problem. The sensor assignment can be affected by construction of optimal control policies $U^*_t = \{u^*(\tau), t_0 \leq \tau \leq \tau\}$, where the state $\mathbf{y}_\alpha(t)$ of the αth sensor is controlled via the dynamics

$$d\boldsymbol{y}_\alpha(t) = a_\alpha(\boldsymbol{y}_\alpha, u_\alpha, t)\, dt + d\boldsymbol{n}_\alpha(t) \tag{18}$$

where $d\boldsymbol{n}_\alpha(t)$ describes the noise process. To be able to solve this problem, we must first determine the transition density $p(\boldsymbol{x}(t), \boldsymbol{y}_\alpha(t), t|\boldsymbol{x}(t_0), t_0, I_{t,t_0}, t_0, U_t)$. This computation, in turn, requires knowledge of the density $p(\boldsymbol{x}(t), t|\boldsymbol{x}(t_0), t_0, I_{t,t_0}, t_0, Y_t)$.

The solution to the second problem is well known. The minimum variance estimate $\hat{\boldsymbol{x}}(t)$ is given by

$$\hat{\boldsymbol{x}}(t) = E\{\boldsymbol{x}(t)|I_{t,t_0}, Y_t\} \tag{19}$$

Therefore, computation of $\hat{\boldsymbol{x}}(t)$ also requires the knowledge of the density $p(\boldsymbol{x}(t), t|\boldsymbol{x}(t_0), t_0, I_{t,t_0}, Y_t)$. The result that follows gives the rules of evolution for the *a posteriori* transition probability density function as already stated here. The derivation permits use of the reports from multiple sensors. The approach also incorporates effects of "positive" information in the same framework. Furthermore, the approach presented here will be able to accommodate joint search/detection/estimation schemes. Finally, multitarget/multisensor systems can also be considered under this formulation. These results are stated without proofs.

Two types of measurements are considered. The continuous measurement is given by

$$dz(t) = h(\boldsymbol{x}(t), t)\, dt + d\boldsymbol{\omega}(t) \tag{20}$$

where $\boldsymbol{w}(t)$ is a Wiener process. The second class of measurements used here is jump processes. The number of detections $N(t)$ will be defined as a jump process with Poisson statistics.

The following theorem gives the equations for the temporal evolution of the transition probability density function for the state $\boldsymbol{x}(t)$ without the use of information $It_{,t_0}$ (43).

Theorem: Let $x(t)$ be a Markov process generated by

$$d\boldsymbol{x}(t) = f(\boldsymbol{x}, t)\,dt + d\beta_{\mathrm{w}} + d\beta_{\mathrm{N}} \tag{21}$$

Let $p = p(\boldsymbol{x}(t), t|x_0, t_0)$ denote the transition probability density function for process $\boldsymbol{x}(t)$. Then p satisfies the partial differential equation

$$\frac{\partial p}{\partial t} = L^{+}(p) \tag{22}$$

where

$$L^{+}(\cdot) = -\sum_{i=1}^{n} \frac{\partial}{\partial x_i}(f_i \cdot) + \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{Q}_{ij}\frac{\partial^2(\cdot)}{\partial x_i \partial x} + \sum_{i=1}^{n} \lambda_i [p_{a_i}{}^{*}\cdot - \cdot] \tag{23}$$

In the preceding equations, $\boldsymbol{Q}_{ij}$ is the covariance matrix associated with the Wiener process $\beta_w(t)$ and λ is the rate parameter associated with the generalized Poisson jump process $\beta_N(t)$,

$$\beta_N(t) = \sum_{i=1}^{x(t)} a_i U(t) \tag{24}$$

with $U(t)$ the unit step function. Also,

$$p_{a_i}{}^{*}p = \int p_{a_i}(u_i - v_i)p(u_1, u_2, \ldots, v_i, \ldots, u_n, t|x(t_0), t_0)\,dv_i \tag{25}$$

where $p_{a_i}(a)$ denotes the density for the random variable a_i.

Equation (23) can be solved analytically for only very specialized cases. Therefore, numerical evaluation of $p(\boldsymbol{x}, t|x_0, t_0)$ will be necessary in using these equations in practice.

The next quantity of interest in the conditional density that describes how the information I_{t,t_0} affects the evolution of the transition density. The preceding unnumbered theorem is to be used to determine the equations for the temporal evolution of $p(\boldsymbol{x}, t|G_{t_0})$ for the following two cases:

1. The measurements are given by $d\boldsymbol{N}(t)$ alone with no accompanying continuous measurements.
2. The measurements are given by $d\boldsymbol{N}(t)$ and $d\boldsymbol{z}(t)$ at time t.

The first case corresponds to a *pure search* policy in which the sensors register detections only without being able to obtain further information about the state $\boldsymbol{x}(t)$. The second case is a more general *surveillance* policy in which the sensors not only perform the search but also are capable of providing tracking information.

Pure Search Strategies

First consider the "pure" search case. There are M sensors surveying the area. During the time interval $[t, t + \Delta t]$, the αth searcher counts the number of detections $dN_\alpha(t)$. The probability that the αth searcher detects the source during $[t, t + \Delta t]$ is given by $\lambda^*_\alpha(\boldsymbol{x}(t), y_\alpha(t))\Delta t$, where $\boldsymbol{x}(t)$ is the state of the source and $\boldsymbol{y}_\alpha(t)$ is the state of the αth sensor at time t. Let

$$\lambda^*(\boldsymbol{x}(t), \boldsymbol{y}(t)) = \begin{bmatrix} \lambda_1^*(\boldsymbol{x}, y_1) \\ \vdots \\ \lambda_M^*(\boldsymbol{x}, y_M) \end{bmatrix} \tag{26}$$

where $\boldsymbol{y}(t)$ is the vector representing the state of all sensors.

Let $d\boldsymbol{N}(t)$ denote the composite report from all searchers defined

$$d\boldsymbol{N}(t) \triangleq \begin{bmatrix} dN_1(t) \\ \vdots \\ dN_M(t) \end{bmatrix} \tag{27}$$

Assuming that the searchers are efficiently deployed, the probability that two sensors will detect the source simultaneously in an interval Δt is infinitesimal and will be ignored. Thus, possible outcomes for $d\boldsymbol{N}(t)$ are

1. $d\boldsymbol{N}(t) = 0$, in which no detections are reported
2. $d\boldsymbol{N}(t) = e_\alpha$, in which the αth sensor reports a detection, and all others report no detections

Theorem: Evolution of density under "pure" search by multiple sensors.

Let $\boldsymbol{x}(t)$ be the vector Markov process defined in Eq. (21) describing the behavior of the signal source. Let the measurement process consist of unit jump process defined by Eq. (27) and with statistics defined by the rate parameter $\lambda^*(\boldsymbol{x}(t), \boldsymbol{y}(t))$ in Eq. (26). Under the assumption that only one sensor detects the source at any given time, the density $p = p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0, G^t_{t_0})$ satisfies (Snyder's equation)

$$\frac{\partial p}{\partial t} = L^{+}(p) + \sum_{\alpha=1}^{M}(\lambda^*_\alpha - E\{\lambda^*_\alpha\})(E\{\lambda^*_\alpha\})^{-1}\left(\frac{dN_\alpha(t)}{dt} - E\{\lambda^*_\alpha\}\right)p \tag{28}$$

where $\lambda^*_\alpha = \lambda^*_\alpha(\boldsymbol{x}(t), \boldsymbol{y}_\alpha(t))$ and the expectation $E\{\lambda^*_\alpha(\boldsymbol{x}(t), \boldsymbol{y}_\alpha(t))\}$ is with respect to the density $p(\boldsymbol{x}(t)|G^t_{t_0})$. The operator $L^{+}(\cdot)$ was defined in Eq. (23).

Search Under Negative Information

An important case of interest is one in which no sensor detects the source over the time interval $[t_0, t_0 + T]$. In this case, $dN(t) \equiv 0$ for $t \in [t_0,, t_0 + T]$ and the conditional density evolves according to

$$\frac{\partial p_0}{\partial t} = L^{+}(p_0) - \sum_{\alpha=1}^{M}[\lambda^*_\alpha(\boldsymbol{x}(t), \boldsymbol{y}_\alpha(t)) - E\{\lambda^*_\alpha(x(t), \boldsymbol{y}_\alpha(t))\}]p_0 \tag{29}$$

This equation is the partial integrodifferential equation that describes the evolution of the transition density of the state of the source when M sensors are searching and have failed to detect the target.

The solution to the preceding partial differential equation, $p(\boldsymbol{x}(t)|G^t_{t_0}, Y_t)$, gives us the fundamental quantity of interest to the fusion and sensor management problems. From an overall systems point of view, the fusion center uses the *a posteriori* density $p(\boldsymbol{x}(t)|G^t_{t_0}, Y_t)$ in many different ways:

1. The optimal sensor control strategies $U^*_t \equiv \{u^*(\tau),\ t_0 \le \tau \le t\}$ are computed to maximize the probability of detection that depends both on $x_t \equiv \{\boldsymbol{x}(\tau),\ t_0 \le \tau \le t\}$ and Y_t.
2. The time evolution of the optimal estimates $\hat{\boldsymbol{x}}(t)$ of the target state $\boldsymbol{x}(t)$ and the associated error covariance matrix $\boldsymbol{P}(t)$ are obtained by integrating with respect to the partial differential equation (29).
3. When a positive report is generated at t_k, Bayes' rule is used to incorporate this information into the fusion process.
4. When classification information is given to the fusion center via cued transformations, it is correlated and used to improve the estimates generated above.
5. The differential equations in Eq. (18) here are similar to Kalman-Bucy filter equations and can be conveniently used in multitarget situations.

Numerical techniques to solve the *nonlinear* partial differential equation in Eq. (29) are not readily available and need to be investigated. Promising approaches are discussed here that provide procedures that are computationally economical, albeit approximate to the second order. First, the differential equations for the mean and covariance are given under the assumption that the density vanishes rapidly as we approach infinity. Solutions to these equations give the all-important conditional mean estimate $\hat{\boldsymbol{x}}(t) = E\{\boldsymbol{x}(t)|G^t_{t_0}\}$ and its error covariance matrix. We assume that these two statistics define with sufficient accuracy the *a posteriori* density as a nearly Gaussian density. For further considerations, the Gaussian form of density is used to derive sensor control strategies.

Joint Search and Track: The Surveillance Policy

The *a posteriori* density functions in Eq. (28) can be feasibly computed for the case in which no positive reports were made by the sensors (i.e., $dN(t) \equiv 0$). Although the equations are valid for situations in which detections are reported by sensors (i.e., $dN(t) \neq 0$), their implementation is not computationally feasible. In the paragraphs that follow, we outline a simplified scheme for enfolding information provided by the sensors reporting a detection. Two approaches are possible to obtain computationally feasible approximations for the first and second moments of $p(\boldsymbol{x}(t)|G^t_{t_0})$. The first approach analytically integrates the partial differential equation. The second approach uses Bayes' rule to compute the conditional density at discrete time points at which detections are available. The resulting equations provide a technique to compute $\hat{\boldsymbol{x}}(t_k)$ and $p(t_k)$ in a recursive manner.

The conditional density $p(\boldsymbol{x}(t)|G^t_{t_0})$ for the state of the target under surveillance then evolves according to Eq. (28).

The conditional mean $\hat{\boldsymbol{x}}(t)$ is defined by

$$\hat{\boldsymbol{x}}(t) = \int_{\Re^n} \boldsymbol{x}(t) p(x(t)|G^t_{t_0})\, d\boldsymbol{x} \tag{30}$$

By integrating Eq. (30) with respect to Eq. (28), the behavior of $\hat{x}(t)$ is given by

$$\begin{aligned}\frac{d\boldsymbol{x}^*}{dt} &= \phi(\boldsymbol{x}^*, t) + \sum_{\alpha=1}^{M} \boldsymbol{P}^*(t)\boldsymbol{D}^T(\boldsymbol{x}^*, t)e_\alpha(\lambda^{*T} e_\alpha)^{-1} \\ &\quad \left(\frac{dN(t)}{dt} - \lambda_\alpha(\boldsymbol{x}^*, \boldsymbol{y}_\alpha)\right)^T e_\alpha\end{aligned} \tag{31}$$

where e_α is the αth coordinate direction in $\Re^n$, $\boldsymbol{P}^*(t)$ is the first-order approximation to the covariance matrix, and $\boldsymbol{D}(\boldsymbol{x}^*, t)$ is the Jacobian matrix for $\lambda(\boldsymbol{x}, \boldsymbol{y})$ with respect to $\boldsymbol{x}$ evaluated at $\boldsymbol{x}^*(t)$. The evolution of $\boldsymbol{P}^*(t)$ is given by

$$\begin{aligned}\frac{d\boldsymbol{P}^*(t)}{dt} &= \boldsymbol{B}(\boldsymbol{x}^*, t) + \sum_{\alpha=1}^{M} \boldsymbol{P}^*(t)\boldsymbol{H}_\alpha(\boldsymbol{x}^* t)\boldsymbol{P}^*(t)e_\alpha^T \frac{dN(t)}{dt} \\ &\quad - \sum_{\alpha=1}^{M} \boldsymbol{P}^*(t)\boldsymbol{E}_\alpha(\boldsymbol{x}^*, t)P^*(t)\end{aligned} \tag{32}$$

where $\boldsymbol{B}(\boldsymbol{x}^*, t) = \boldsymbol{A}(\boldsymbol{x}^*, t)\boldsymbol{P}^*(t) + \boldsymbol{P}^*(t)\boldsymbol{A}^T(\boldsymbol{x}^*, t) + \boldsymbol{Q}(t)$, $\boldsymbol{A}(\boldsymbol{x}^*, t)$ is the Jacobian matrix for $\phi(\boldsymbol{x}, t)$, $\boldsymbol{E}_\alpha(\boldsymbol{x}, t)$ is the Hessian for $\lambda_\alpha(\boldsymbol{x}, t)$ and $\boldsymbol{H}_\alpha(\boldsymbol{x}, t)$ is the Hessian for $\ln[\lambda_\alpha(\boldsymbol{x}, t)]$.

The important case, in which there are no detections in a given interval, is much easier to solve. Equations for this case are obtained by setting $dN(t)/dt \equiv 0$ over the interval $[t_k, t_{k+1}]$, where t_k denotes the sequence of arrival times for the detections. At t_k, where a detection is reported, the density is updated using Bayes' rule. Although this approach seems computationally complex, the differential equations to be solved for $\boldsymbol{x}^*_0(t)$ and $\boldsymbol{P}^*_0(t)$ (i.e., no detection case) are quite similar to the continuous time Kalman filtering equations, and solutions are computationally feasible. The algorithm is described in what follows.

Let t_k be the sequence of times at which detections are reported by any one of the sensors. For notational simplicity, denote $G^k \triangleq G^{t_k}_{t_0}$ and

$$p(\boldsymbol{x}_k|G_k) \triangleq p(\boldsymbol{x}(t_k)|G^{t_k}_{t_0})$$

Assume that at time t_{k-1} the fusion center has computed $p(\boldsymbol{x}_{k-1}|G^{k-1})$. Also, during the interval $[t_{k-1}, t_k]$ some of the sensors detect the source. Let $\overline{N}_k$ be the index set for sensors that did not report detections. The following procedure provides a general technique for computing $p(\boldsymbol{x}_k|G^k)$ using $p(\boldsymbol{x}_{k-1}|G^{k-1})$ and the information provided by the sensors during $[t_{k-1}, t_k]$. Denote by $\overline{G}^k$ the information provided by negative reports from sensors in the index set $\overline{N}_k$.

Step 0. Initialize $p(\boldsymbol{x}_0|G^0)$ using *a priori* information about the target.

For $k = 1, 2, \ldots,$

Step 1. Using Eq. (31), compute $p_0(\boldsymbol{x}_k|\overline{G}_k)$ using $p(\boldsymbol{x}_{k-1}|G^{k-1})$ as the initial density.

Step 2. Compute $p(\boldsymbol{x}_k|G^k)$ using Bayes' rule and $p(\boldsymbol{x}_k|\overline{G}_k)$.

In the preceding algorithm, Step 1 uses either the solution to the partial differential equation (31) or the recursive formulation of the representation for $p(\boldsymbol{x}_k|\overline{G}^k)$ given in Eq. (29). An alternative approach for computing $p(\boldsymbol{x}_k|\overline{G}^k)$ is to determine $\hat{\boldsymbol{x}}_k(\overline{G}_k)$ and $\boldsymbol{P}_k(\overline{G}_k)$ using differential equations (31) and (32) and an approximate Gaussian form

$$p(\boldsymbol{x}_k|\overline{G}_k) = N(\boldsymbol{x}_k|\hat{\boldsymbol{x}}_k(\overline{G}_k), \boldsymbol{P}_k(\overline{G}_k))$$

For computations in Step 2, two distinct cases must be considered. To simplify the analysis, assume that only one sensor (αth) reports a positive detection ($D_k = 1$) during $[t_{k-1}, t_k]$. Then, one of two things can result:

1. The αth sensor does not provide any further information about the source and $G_k = (\overline{G}^k, D_k = 1)$.
2. The αth sensor provides a measurement $\boldsymbol{z}_k$ at time t_k. Assume that this measurement has the form $\boldsymbol{z}_k = h_k(\boldsymbol{x}_k, \boldsymbol{y}_k) + \boldsymbol{v}_k$. In this case,

$$G^k = (\overline{G}^k, D_k = 1, \boldsymbol{z}_k) \tag{33}$$

In both cases, Bayes' rule is applied to compute $p(\boldsymbol{x}_k|G^k)$ from $p(\boldsymbol{x}_k|\overline{G}^k)$.

Having computed the effects of a sensor report on the location density of a signal source, the next step is to use this information for efficiently assigning several sensors.

Sensor Control

Consider the surveillance problem in which the *a priori* target state at time t_0 is described through the density function $p(\boldsymbol{x}|G_0)$, where $\boldsymbol{x}$ denotes the target state and G_0 denotes all available information up to t_0. The problem is to determine the part of the state space, denoted by S, to be closely watched by a single sensor at subsequent time instants t_k. We define

$$t_k = t_{k-1} + \Delta t_k \quad \text{for} \quad k = 1, 2, \ldots, N \tag{34}$$

where Δt_k is the time interval of search at the kth stage of the search. In general, the surveillance space S is n-dimensional. For the sake of practicality, however, we focus attention on the two-dimensional space described by the latitude-longitude of the target. The analysis presented here is general enough to include first and higher derivatives of the position as well as other parameters of interest. The optimal sensor control problem is that of computing the sensor search plan $\{u(\boldsymbol{x}, \tau)\ \tau \in (t_0, t\}\}$ that will maximize the detection probability (or some other suitable criterion) resulting from a search plan.

The surveillance space $S \subseteq$ is divided into M discrete cells; the ith cell is denoted Λ_i. The simplest sensor control problem is one in which only one sensor is available and during any interval $I_k \triangleq [t_k, t_{k+1}]$, $k = 0, 1, \ldots, N - 1$, the sensor can search only one of the M cells that collectively constitute the surveillance space S. Therefore, at t_k, a decision is to be made as to which one of the M cells the sensor ought to search. The measure of performance for evaluating different cell assignments for search will be the target state uncertainty. A computable measure of this uncertainty is the average entropy of the location density after a report from the sensor. It should be noted here that when a detection is reported in a cell, the probability mass for the location density peaks in the surveillance region around that cell. If the sensor scans the cell and reports no detection, then the probability mass around that cell depletes and the mass is spread throughout the remainder of the surveillance region. This contraction and spreading of the probability mass is reflected in the average entropy of the probability distribution after a report. As this distribution starts peaking in a particular area, the entropy of the distribution decreases. As the probability distribution spreads, the average entropy is increased. This effect of detection performance on the entropy is intuitively appealing. Because the target is dynamic, if a detection is not made it gives the target time to move around in the surveillance region and, thus, increases uncertainty. On the other hand, a detection in a particular cell localizes the target and reduces uncertainty.

The optimization process involves three steps:

1. Computation of the posterior density after a report
2. Computation of the average entropy of the posterior density
3. Assignment of the sensor to a cell in the surveillance region

The methods discussed in the text provide the means to accomplish Step 1. Step 3 is trivially accomplished once the average entropy of the posterior density function is computed. A recursive control policy for this case and computational results are given in Ref. 43.

BIBLIOGRAPHY

1. S. Blackman, *Multiple-Target Tracking with Radar Applications,* Norwood, MA: Artech House, 1986.
2. E. Waltz and J. Llinas, *Multisensor Data Fusion,* Norwood, MA: Artech House, 1990.
3. R. T. Antony, *Principles of Data Fusion,* Norwood, MA: Artech House, 1995.
4. Y. Bar-Shalom, *Multitarget-Multisensor Tracking: Advanced Applications,* volume I, Norwood, MA: Artech House, 1990
5. Y. Bar-Shalom, *Multitarget-Multisensor Tracking: Applications and Advances,* volume II, Norwood, MA: Artech House, 1992.
6. Y. Bar-Shalom and X. R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques,* Storrs, CT: YBS Publishing, 1995.
7. D. B. Reid, An algorithm for tracking multiple targets, *IEEE Trans. Autom. Control,* **34**: 843–854, 1976.
8. T. Kurien, Issues in the design of practical multitarget tracking algorithms, in Bar-Shalom (ed.), *Multitarget-Multisensor Tracking: Advanced Applications,* Norwood, MA: Artech House, 1990, pp. 43–83.
9. A. B. Poore, Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking, *Computat. Optim. Appl.,* **3**: 27–57, 1994.
10. J. J. Stein and S. S. Blackman, Generalized correlation of multitarget data, *IEEE Trans. Aerosp. Electron. Syst.,* **AES-11**: 1207–1217, 1975.
11. H. W. Sorenson, *Parameter Estimation,* New York: Marcel Dekker, Inc., 1980.
12. P. E. Caines, *Linear Stochastic Systems,* New York: Wiley, 1988.
13. S. C. Nardone and V. J. Aidala, Observability criteria for bearings-only target motion analysis, *IEEE Trans. Aerosp. Electron. Syst.,* **AES-171**: 262–266, 1981.

14. S. E. Hammel and V. J. Aidala, Observability requirements for three-dimensional tracking via angle measurements, *IEEE Trans. Aerosp. Electron. Syst.,* **AES-21**: 200–207, 1985.
15. A. G. Lindren and K. F. Gong, Position and velocity estimation via bearing observations, *IEEE Trans. Aerosp. Electron. Syst.,* **AES-14**: 564–577, 1978.
16. V. J. Aidala, Kalman filter behavior in bearings-only tracking applications, *IEEE Trans. Aerosp. Electron. Syst.,* **AES-15**: 29–39, 1979.
17. V. J. Aidala and S. E. Hammel, Utilization of modified polar coordinates for bearings-only tracking, *IEEE Trans. Autom. Control,* **AC-28**: 283–294, 1983.
18. H. W. Sorenson and D. L. Alspach, Recursive Bayesian estimation using Gaussian sums, *Automatica,* **7**: 465–479, 1971.
19. D. Alspach, A Gaussian sum approach to the multitarget identification-tracking problem, *Automatica,* **11**:285–296, 1975.
20. D. P. Bertsekas and D. A. Castanon, A forward/reverse auction algorithm for asymmetric assignment problems, *Computat. Optim. Appl.,* **1**: 277–297, 1992.
21. A. B. Poore, Multidimensional assignments and multitarget tracking, in *Partitioning Data Sets,* I. J. Cox, P. Hansen, B.Julesz (eds.), vol. 19, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science,* Providence, RI: American Mathematical Soc., 1995, pp. 169–198.
22. A. B. Poore and A. J. Robertson III, A new Lagrangian relaxation based algorithm for a class of multidimensional assignment problems, *Computat. Optim. Appl.,* **8** (2): 129–150, 1997.
23. A. B. Poore and N. Rijavec, A Lagrangian relaxation algorithm for multidimensional assignment problems arising from multitarget tracking, *SIAM J. Optim.,* **3**: 544–563, 1993.
24. K. R. Pattipati et al., A new relaxation algorithm and passive sensor data association, *IEEE Trans. Autom. Control,* **AES-37**: 198–213, 1992.

24a. T. Kirubarajan and Y. Bar-Shalom, Low observable target motion analysis using amplitude information, *IEEE Trans. Aerosp. Electron. Syst.,* **AES-32**: 1367–1384, 1996.

25. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,* Palo Alto, CA: Morgan Kaufmann Publ., 1988.
26. B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence,* Englewood Cliffs, NJ: Prentice-Hall, 1992.
27. G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications,* Upper Saddle River, NJ: Prentice-Hall, 1995.
28. D. Driankov, H. Hellendoorn, and M. Reinfrank, *An Introduction to Fuzzy Control,* Berlin: Spring-Verlag, 1993.
29. J. J. Kruger and I. S. Shaw, A fuzzy learning system emulating a human tracking operator, *Proc. 1st Int. Symp. Uncertainty Modeling Analysis,* College Park, MD, 1990, pp. 25–28.
30. P. J. Pacini and B. Kosko, Adaptive fuzzy systems for target tracking, *J. Intell. Syst. Eng.,* **1** (1): 3–21, 1992.
31. C. G. Moore, C. J. Harris, and E. Rogers, Utilizing fuzzy models in the design of estimators and predictors: an agile target tracking example, *Proc. 2nd IEEE Int. Conf. Fuzzy Syst.,* **2**, March 1991, pp. 679–684.
32. C.-W. Tao, W. E. Thompson, and J. S. Taur, A fuzzy logic approach to multidimensional target tracking, *Proc. 2nd IEEE Int. Conf. Fuzzy Syst.,* San Francisco, CA, **2**: 1991, pp. 1350–1355.
33. Y. H. Lho and J. H. Painter, A fuzzy-tuned adaptive Kalman filter, *Proc. 3rd Int. Conf. Ind. Fuzzy Control Intell. Syst.,* Houston, TX, 1993, pp. 144–148.
34. C.-W. Tao et al., An estimator based on fuzzy if-then rules for the multisensor multidimensional multitarget tracking problem, *Proc. 3rd Conf. Fuzzy Syst.,* **3**: 1994, pp. 1543–1548.
35. R. Priebe and R. Jones, Fuzzy logic approach to multitarget tracking in clutter, *SPIE Acquisition, Tracking, and Pointing V,* **1482**: 1991, pp. 265–274.
36. F. A. Watkins, "Fuzzy Engineering," Ph.D. thesis, Univ. California, Irvine, Dept. Electr. Comput. Eng., June 1994.
37. R. N. Lobbia, Sensor fusion implementation with neural networks and fuzzy logic, ORINCON Tech. Rep. OCR 95-4155-U-0371, October 1995.
38. L. Hong and G-J. Wang, personal communication, 1992.
39. R. N. Lobbia and S. C. Stubberud, Autonomous neural control of space platforms, ORINCON Tech. Rep. OCR 94-4050-U-0037, February 1994.
40. S. C. Stubberud, R. N. Lobbia, and M. Owen, An adaptive extended Kalman filter using artificial neural networks, *Proc. 34th IEEE Conf. Decision Control,* New Orleans, LA: 1995.
41. S. C. Stubberud, R. N. Lobbia, and M. Owen, Adaptive state estimation using artificial neural networks, *Proc. ANNIE'95,* St. Louis, MO, 1995.
42. S. Singhal and L. Wu, Training multilayer perceptrons with the extended Kalman algorithm, in *Advances in Neural Processing Systems I,* D. S. Touretsky (ed.), Palo Alto, CA: Morgan Kaufmann, 1989, pp. 133–140.
43. V. Samant, Estimation and control approaches to sensor control in C3S systems, *Proc. 5th MIT/ONR Workshop C3 Syst.,* Monterey, CA, 1982.

VIVEK SAMANT
DALE KLAMER
ORINCON Corporation

SONICS IN GEOPHYSICAL PROSPECTING. See GEOPHYSICAL PROSPECTING USING SONICS AND ULTRASONICS.

SONOLUMINESCENCE AND SONOCHEMISTRY, PHYSICAL MECHANISMS AND CHEMICAL EFFECTS. See ULTRASONIC PHYSICAL MECHANISMS AND CHEMICAL EFFECTS.

SOUND-LEVEL INSTRUMENTS. See LEVEL METERS.

SOUND LOUDNESS. See PSYCHOACOUSTICS.

SOUND MASKING. See PSYCHOACOUSTICS.

SOUND PRESSURE. See ACOUSTIC VARIABLES MEASUREMENT.

SOUND, PRODUCTION. See MUSICAL INSTRUMENTS.

SOUND WAVES, UNDERWATER. See UNDERWATER ULTRASOUND.