

# Tutorial on Multisensor Management and Fusion Algorithms for Target Tracking

Michael K. Kalandros, Lidija Trailović, Lucy Y. Pao, and Yaakov Bar-Shalom

**Abstract**—This paper provides an introduction to sensor fusion techniques for target tracking. It presents an overview of common filtering techniques that are effective for moving targets as well as methods of overcoming problems specific to target tracking, such as measurement-to-track association and sensor registration. The computational demand of such algorithms is discussed and various practices, including distributed processing of target tracks and sensor management, are proposed to help reduce this demand. Final comments include a discussion of applications and implementation issues specific to the presented scenarios.

## I. INTRODUCTION

Target tracking is the process of maintaining state estimates of one or several objects over a period of time. These objects can be aircraft, ships, or ground-based targets. Mobile robots can track the location of landmarks in their environment to maintain an accurate estimate of their location, as well. Target tracking algorithms are basically state estimation algorithms, where the estimate of the state is corrected by measurements from various sensors, which can include radar, sonar, and CCD cameras, to name a few. An illustration of tracking multiple aerial targets using sensors such as radars is presented in Figure 1. The use of multiple sensors can dramatically improve tracking accuracy in a process known as sensor fusion. Section II discusses the extension of common state estimation and target tracking algorithms, such as the Kalman filter [9], to include the fusion of data from multiple sensors based on a centralized processing architecture as shown in Figure 2.

One of the difficulties that sets target tracking apart from other estimation tasks is the uncertainty in measurement origin. Typically in state estimation tasks, the measurement is clearly linked to the object about which state information is desired. In many tracking environments, however, measurements are received from multiple objects, and it is not known which measurement originated from which object. In these environments (radar-based tracking, for example) there is no clear linkage of measurement to object,

This work was partially supported by the Office of Naval Research (Grant N00014-02-1-0136). The authors thank Weerawat Khawsuk for his comments on Section 5, and Richard M. Powers for providing the data for Figures 5, 6, and 7.

M. K. Kalandros is with Air Defense Systems Department, The Johns Hopkins Applied Physics Lab, Johns Hopkins Road, Laurel, MD 20723-6099, michael.kalandros@jhuapl.edu.

L. Trailović and L. Y. Pao are with the Department of Electrical and Computer Engineering, University of Colorado, Boulder, CO 80309-0425, lidija.trailovic@colorado.edu, pao@colorado.edu.

Y. Bar-Shalom is with the Department of Electrical and Systems Engineering, University of Connecticut, Storrs, CT 06269-1157, ybs@engr.uconn.edu.

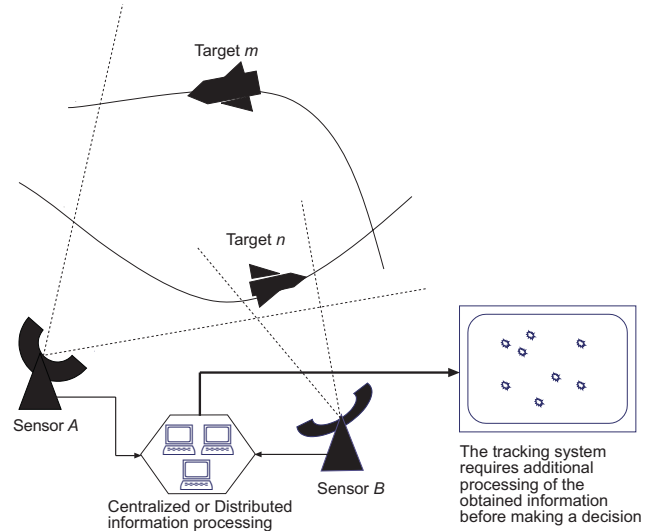


Fig. 1. An illustration of the sensor fusion idea: the radars provide measurements of the surveillance region and the processing units (centralized or distributed) gather data, perform a sensor fusion algorithm, and determine positions of targets. Sensor fusion algorithms are capable of combining information from diverse sensing equipment, and improve tracking performance, but at a cost of increased computational complexity.

requiring *data association* algorithms to associate measurements to target tracks. Data association is further complicated by the possible presence of non-target or “clutter” measurements (detections or returns from nearby objects, clouds, electromagnetic interference, acoustic anomalies, false alarms, *etc.*) that must be identified as not originating with any targets of interest. The resulting uncertainty about measurement origin decreases the accuracy of the state estimate of each track and can ultimately lead to track loss. One approach to data association is known as the Nearest Neighbor (NN) filter, which simply selects the measurement that is closest to the predicted measurement, discarding all the others ([8], p. 123). Another approach is Probabilistic Data Association (PDA), which creates a weighted average of all the received measurements, based on the likelihood that each measurement came from the target of interest ([8], Section 3.4). Sensor fusion techniques for tracking environments requiring data association are discussed in Section III.

Another factor that affects the accuracy of sensor fusion is imperfect knowledge of the location of each sensor, leading to errors common to each measurement from that sensor. Techniques to reduce these errors, discussed in Section IV, generally involve augmenting the state of the target track to include estimates of the sensor’s relative location.

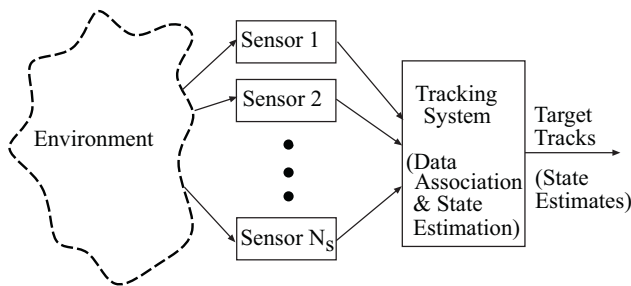


Fig. 2. Tracking system using a centralized processing architecture. It includes a set of sensors that observe the environment, including the targets to be tracked, and a central system for processing the outputs of those sensors.

While the fusion of multiple sensor measurements can result in dramatically increased tracking performance, it often leads to large increases in communication load and computational complexity. One method of implementing such algorithms is to distribute the tracking task across multiple processors. For example, each sensor could have its own processor that uses single-sensor tracking algorithms. The tracks from each processor are either shared among the different processors or sent to a “global” processor for fusion. This approach has its own drawbacks, most notable of which is that while measurements of a target from different sensors are generally uncorrelated, the target tracks created from the measurements are correlated, requiring additional processing. However, since the number of target tracks is generally much lower than the number of individual measurements, the complexity seen by each processor will still be less than that required for a single centralized processor system. Distributed and decentralized tracking are discussed in Section V.

As the number of targets and sensors increases, tracking systems can very quickly become overloaded by the incoming data. Furthermore, as the number of available sensors and sensor modes increases, it is easy to overwhelm human operators, such as fighter pilots, as well. Sensor manager systems that balance tracking performance with system resources are often required to control the flow of information. Although functions controlled by sensor managers include a wide array of sensing activities such as the selection of sensor modes or scanning an area for unknown targets, we primarily overview methods for determining how best to task sensors to illuminate specific targets in Section VI. By setting an allowable threshold on track accuracy, sensors can be assigned by the needs of each target track, instead of attempting to apply all sensors to all targets. Alternatively, sensors can be selected in such a way that their effectiveness is maximized, thus deriving the most information from the available sensor resources.

In principle, fusion of multisensor data provides significant advantages over single source data. In addition to the statistical advantage gained by combining same-source data (*e.g.*, obtaining an improved estimate of a physical phenomenon through redundant observations), the use of multiple types of sensors may increase the accuracy of the

observation. For example, let us consider a moving object such as an aircraft, observed by both a pulsed radar and an infrared imaging sensor. The radar provides accuracy in determining the aircraft’s range, but a limited ability to determine the angular direction of the aircraft. On the other hand, the infrared imaging sensor has accuracy in determining the aircraft’s angular direction, but is unable to measure range. If these two observations can be correctly associated, the combination of the two sensor data would provide improved estimates of the location (better than either of the two independent sensors), but at a cost of additional computation.

Computational complexity studies of the tracking algorithms are numerous and can be categorized as studies of the filter complexity and studies of data association complexity. For example, comparison of different filtering techniques in terms of handling nonlinearities were discussed in [2], [62], [63], similar tracking scenarios were analyzed in [26], [28], [36], [74], and comparison of several data association algorithms were addressed in [60]. More on computational issues in tracking algorithms will be presented in Section VII.

Multisensor fusion was designed to combine sensor information from a variety of sensors and to improve target state estimates (over those resulting from using a single sensor) in a surveillance region, or in other words, to resolve uncertainty in retrieved data sets. Significant activity has taken place in the research and development of sensor and data fusion systems for military applications such as automatic target recognition, mission-critical systems (in battlefield assessment), and for smart weapons. More recently, applications such as integration of intelligent systems (automated plant management, computer vision, vehicle teleoperation, intelligent robot navigation, *etc.*), atmospheric sciences data modeling (*e.g.*, handling large state spaces in remote sensing), medical diagnosis (*e.g.*, imaging), and smart buildings are beginning to apply methods that have originated in multisensor fusion. A number of projects of interdisciplinary nature for civilian applications have been carried out — Distributed Space Systems Technology, Earth Science Technology, to name a few — where some form of uncertainty is built in the acquired data sets. Implementation of multisensor fusion systems has become practical in the last decade due to the advances in sensor manufacturing, signal processing algorithms, VLSI technology, and high performance computing and communication [29].

It is worth noting the difference between *sensor data fusion* and *general data fusion* (or frequently, *information fusion*), according to terminology suggested in [29]. *Multisensor fusion* implies a process which generally employs both correlation and fusion processes to transform sensor measurements into updated states and covariances for entity tracking (also known in the literature as *correlator-tracker* or *level 1 fusion*). *Information fusion* refers to acquiring, processing, and intelligently combining information gathered by various knowledge sources and sensors to provide a better understanding of the phenomenon under consideration. It implies the use of multisensor fusion as an

integral part of a more complex process. Information fusion is capable of handling data diverse in nature and origin, requires access to a variety of resources (e.g., data/knowledge base), and can be described as a process by which the tracked entities are associated with environmental, doctrinal, and performance constraints, or presents a structured multi-perspective assessment of entities and trends (*situation assessment* and *threat assessment*, or *level 2* and *level 3 fusion*). Techniques for information fusion are complex and drawn from a wide range of areas including artificial intelligence, pattern recognition, statistical estimation, etc. Though the terms *sensor fusion* and *information fusion* may be used interchangeably in the literature, issues related to information fusion as described above are beyond the scope of this tutorial paper.

## II. FILTERING

From a Bayesian perspective, the general tracking problem is to recursively calculate some degree of belief that the state  $\mathbf{x}(k)$  at time  $k$  has some value given the data  $\mathbf{z}(1:k)$  up to time  $k$ . Thus, it is required to construct the pdf  $p(\mathbf{x}(k)|\mathbf{z}(1:k))$ . It is assumed that the initial pdf  $p(\mathbf{x}(0)|\mathbf{z}(0)) \equiv p(\mathbf{x}(0))$  of the state vector, also known as the prior, is available ( $\mathbf{z}(0)$  being the set of no measurements). In principle, the pdf  $p(\mathbf{x}(k)|\mathbf{z}(1:k))$  may be obtained recursively in two stages: prediction and update. In the update stage the measurement  $\mathbf{z}(k)$  is used to modify the prior density to obtain the required posterior density of the current state.

To define the general problem of tracking, let us consider the evolution of the state-space model of a target defined by a sequence  $\{\mathbf{x}(k)\}$ ,

$$\mathbf{x}(k) = \mathbf{f}_k(\mathbf{x}(k-1), \mathbf{w}(k-1)), \quad (1)$$

where  $\mathbf{f}_k(\cdot)$  is a possibly time-varying nonlinear function of the state  $\mathbf{x}(k)$ ,  $\mathbf{w}(k)$  is an i.i.d. process noise sequence, and  $k$  is the discrete time index. The tracking objective is to recursively estimate  $\mathbf{x}(k)$  based on measurements from sensor  $j$

$$\mathbf{z}(k) = \mathbf{h}_k(\mathbf{x}(k), \mathbf{v}(k)), \quad (2)$$

where  $\mathbf{h}_k(\cdot)$  is a possibly time-varying nonlinear function of the state, and  $\mathbf{v}(k)$  is an i.i.d. measurement noise sequence. In particular, tracking is the calculation of filtered estimates of  $\mathbf{x}(k)$  based on the set of all available measurements  $\mathbf{z}(1:k) = \{\mathbf{z}(i), i = 1, \dots, k\}$  up to time  $k$ .

The recursive relations for prediction and update form the basis for the optimal Bayesian solution [3] but in general it can not be determined analytically. Solutions exist in a restrictive set of cases. In the special case when the state and measurement equations are linear, the noise is Gaussian, and in the absence of clutter, the optimum analytical solution to the problem of recursively estimating the states  $\mathbf{x}(k)$  based on the measurements  $\mathbf{z}(1:k)$  is given by the Kalman filter (KF).

The Kalman filter is based on the following models of the target and measurement systems ([7], Chap. 5):

$$\mathbf{x}(k) = \mathbf{F}\mathbf{x}(k-1) + \mathbf{w}(k-1) \quad (3)$$

$$\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k) \quad (4)$$

where  $\mathbf{x}(k)$  is the current state of the target;  $\mathbf{F}$ ,  $\mathbf{H}$  are known system matrices;  $\mathbf{z}(k)$  is the measurement of the target;  $\mathbf{w}(k)$  is a variable representing process noise or higher-order motion not modeled by  $\mathbf{F}$ ; and  $\mathbf{v}(k)$  is a variable representing measurement noise in the sensor. Both  $\mathbf{w}(k)$  and  $\mathbf{v}(k)$  are assumed to have zero-mean, white, Gaussian probability distributions.

Since  $\mathbf{w}(k)$  and  $\mathbf{v}(k)$  are zero-mean noise processes, the target states and measurements in the next time interval can be predicted by

$$\hat{\mathbf{x}}(k|k-1) = \mathbf{F}\hat{\mathbf{x}}(k-1|k-1) \quad (5)$$

$$\hat{\mathbf{z}}(k|k-1) = \mathbf{H}\hat{\mathbf{x}}(k|k-1) \quad (6)$$

where  $\hat{\mathbf{x}}(k|k-1)$  is the expected value of  $\mathbf{x}(k)$ ,  $E[\mathbf{x}(k)]$ , given data up to the time period  $k-1$  and  $\hat{\mathbf{z}}(k|k-1)$  is the expected value of the measurement at time  $k$ . The quantity  $\nu(k) = \mathbf{z}(k) - \hat{\mathbf{z}}(k|k-1)$  is known as the innovation. The covariance of the state and the innovation predictions are  $\mathbf{P}$  and  $\mathbf{S}$ , respectively:

$$\begin{aligned} \mathbf{P}(k|k-1) &= E[(\mathbf{x}(k) - \hat{\mathbf{x}}(k|k-1))(\mathbf{x}(k) - \hat{\mathbf{x}}(k|k-1))'] \\ &= \mathbf{F}\mathbf{P}(k-1|k-1)\mathbf{F}' + \mathbf{Q}(k-1) \end{aligned} \quad (7)$$

$$\begin{aligned} \mathbf{S}(k) &= E[(\mathbf{z}(k) - \hat{\mathbf{z}}(k))(\mathbf{z}(k) - \hat{\mathbf{z}}(k))'] \\ &= \mathbf{H}\mathbf{P}(k|k-1)\mathbf{H}' + \mathbf{R}(k) \end{aligned} \quad (8)$$

where  $\mathbf{Q}(k)$  is the process noise covariance and  $\mathbf{R}(k)$  is the measurement noise covariance. The state estimate is then updated with the new measurement:

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(k)(\mathbf{z}(k) - \mathbf{H}\hat{\mathbf{x}}(k|k-1)) \quad (9)$$

where  $\mathbf{K}(k)$  is the Kalman gain:

$$\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{H}'\mathbf{S}^{-1}(k). \quad (10)$$

The state covariance is updated by

$$\mathbf{P}(k|k) = (\mathbf{I} - \mathbf{K}(k)\mathbf{H})\mathbf{P}(k|k-1). \quad (11)$$

The Kalman filter, as well as the other, more advanced algorithms in this section, can be extended to include inputs from multiple sensors per scan in either a parallel or sequential fashion [81]. In the parallel implementation, the measurement vectors  $\mathbf{z}_s(k) = \mathbf{H}_s\mathbf{x}(k)$  are stacked into a single vector  $\mathbf{Z}(k) = [\mathbf{z}_1(k), \mathbf{z}_2(k), \dots, \mathbf{z}_{N_s}(k)]$ , where  $N_s$  is the total number of sensors and  $s$  is the sensor index, and then processed as a single sensor as described above. However, extending the size of the measurement vector can dramatically increase the computational complexity of the algorithm, particularly in the calculation of the inverse of the innovation covariance matrix for the Kalman gain in (10). Because of this, the parallel approach to multisensor fusion is rarely used. In the sequential implementation a different KF is run for each sensor, with the *a posteriori* state

estimate and covariance from each sensor's KF becoming the *a priori* state estimate and covariance for the following sensor's KF.

A central operation performed in the KF is the propagation of a Gaussian random variable through the system dynamic equations (5) and (6). However, many real-world systems, such as target tracking problems with clutter and/or maneuvering targets, involve elements of nonlinearity and non-Gaussianity, which preclude exact analytic solution.

Resolving nonlinearities in tracking problems motivated the development of a number of suboptimal filtering algorithms. The extended Kalman filter (reviewed in [2], [78]) has become a standard technique used in a number of nonlinear estimation and machine learning applications. These include estimating the state of nonlinear dynamic systems, estimating parameters for nonlinear system identification (*e.g.*, learning the weights of a neural network), and dual estimation (*e.g.*, the expectation maximization (EM) algorithm) where both states and parameters are estimated simultaneously. In the extended Kalman filter (EKF), the states are propagated using their nonlinear descriptions in (1) and (2) without contributions from the noise components,

$$\hat{\mathbf{x}}(k) = \mathbf{f}_k(\mathbf{x}(k-1)) \quad (12)$$

$$\hat{\mathbf{z}}(k) = \mathbf{h}_k(\mathbf{x}(k)), \quad (13)$$

and then the state distribution is approximated by a Gaussian random variable. The state estimate is then propagated analytically through a first-order linearization of the nonlinear system by replacing the state transition matrix  $\mathbf{F}$  and the measurement matrix  $\mathbf{H}$  with the partial derivatives of the nonlinear functions of the state and measurement, respectively,

$$\mathbf{F}(k) = \left. \frac{\partial(\mathbf{x}(k))}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k)}, \quad (14)$$

$$\mathbf{H}(k) = \left. \frac{\partial \mathbf{h}_k(\mathbf{x}(k))}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k)}. \quad (15)$$

The new  $\mathbf{F}$  and  $\mathbf{H}$  matrices are used in the rest of the KF equations, which are unchanged. If the true density is non-Gaussian (*e.g.*, bimodal or heavily skewed), this can introduce large errors in the true posterior mean and covariance of the transformed Gaussian random variable, which may lead to sub-optimal performance and sometimes divergence of the EKF. The unscented Kalman filter (UKF) addresses this problem by using a deterministic sampling approach. The UKF has been proposed [32], [33] and adopted in recent years as an alternative to the EKF. The state distribution is again approximated by a Gaussian random variable, but is now represented using a minimal set of carefully chosen sample points (called Sigma points) instead of approximating the nonlinear system dynamics. These sample points capture the true mean and covariance completely, and when propagated through the true nonlinear system, capture the posterior mean and covariance accurately to the third order (of the Taylor series expansion)

for any nonlinearity. Methods that demand approximation and/or linearization (as does the EKF) modify the system model itself in order to solve the filtering problem. The EKF achieves only first-order accuracy, but interestingly, the computational complexity of the UKF is the same order as that of the EKF.

The first step in the UKF is to sample the prior state distribution, that is, generate the Sigma points:

$$\begin{aligned} \chi_{0,k-1} &= \hat{\mathbf{x}}_{k-1} \\ W_0 &= \frac{\kappa}{(n+\kappa)} \\ \chi_{i,k-1} &= \hat{\mathbf{x}}_{k-1} + \left( \sqrt{(n+\kappa)\mathbf{P}_{k-1}} \right)_i \\ W_i &= \frac{1}{2(n+\kappa)} \\ \chi_{i+n,k-1} &= \hat{\mathbf{x}}_{k-1} - \left( \sqrt{(n+\kappa)\mathbf{P}_{k-1}} \right)_i \\ W_{i+n} &= \frac{1}{2(n+\kappa)} \end{aligned} \quad (16)$$

where  $i = 1, 2, \dots, n$ , and  $n$  is the state dimension. Each Sigma point  $(\chi_{i,k-1})$  has an associated weight  $(W_{i,k-1})$ . The parameter  $\kappa$  is a design parameter of the UKF which affects the the distribution of the Sigma points. After the prior distribution is sampled to generate the Sigma points, they are propagated through the nonlinear model. This is the *prediction* step of the UKF:

$$\chi_{i,k} = f(\chi_{i,k}, u_{k-1}, (k-1)) \quad (17)$$

$$\hat{\mathbf{x}}_{i,k}^- = \sum_{i=0}^{2n} W_i \chi_{i,k} \quad (18)$$

$$\mathbf{P}_k^- = \sum_{i=0}^{2n} W_i (\chi_{i,k} - \hat{\mathbf{x}}(k)^-) (\chi_{i,k} - \hat{\mathbf{x}}(k)^-)' \quad (19)$$

In the UKF, the prediction step also requires determining the output distribution by propagating the Sigma points through the nonlinear output equation. Denoting the output Sigma points as  $\chi_{i,k}^y$ , the output Sigma points and the distribution are

$$\chi_{i,k}^y = h(\chi_{i,k-1}^y, u_{k-1}, (k-1)) \quad (20)$$

$$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2n} W_i \chi_{i,k}^y \quad (21)$$

$$\mathbf{P}_{yy,k}^- = \sum_{i=0}^{2n} W_i (\chi_{i,k}^y - \hat{\mathbf{y}}_k^-) (\chi_{i,k}^y - \hat{\mathbf{y}}_k^-)' \quad (22)$$

$$\mathbf{P}_{xy,k}^- = \sum_{i=0}^{2n} W_i (\chi_{i,k} - \hat{\mathbf{x}}_k^-) (\chi_{i,k}^y - \hat{\mathbf{y}}_k^-)' \quad (23)$$

In the *correction* step, an alternative Kalman gain is calculated and used to update the *a priori* estimate with the measurement,

$$\mathbf{K}_k = \mathbf{P}_{xy,k}^- \left( \mathbf{P}_{yy,k}^- + \mathbf{R}_k \right)^{-1}. \quad (24)$$

The state and covariance updates are calculated as in the standard Kalman filter, using the alternative gain computation.

Another approach when resolving the nonlinear and non-Gaussian nature of tracking is using Sequential Monte Carlo methods that have evolved from the general problem of estimation in nonlinear, non-Gaussian systems such as particle or bootstrap filters. In many estimation and tracking application areas it is important to include both nonlinear or non-Gaussian elements in order to accurately model the dynamics of physical systems. To reduce storage costs and allow for rapid adaptation to changing signal characteristics, it is required to process data on-line in a sequential manner. Particle filters [2], [22] are sequential Monte Carlo methods based on point mass (or “particle”) representations of probability densities, which can be applied to any (possibly nonlinear and non-Gaussian) system model, and which generalize traditional Kalman filtering methods. Because of the highly nonlinear dynamics and measurement models, and non-Gaussian distributions in ground moving target scenarios, particle filters have been applied for littoral tracking (targets on land or in the sea near the boundary) and yielded good tracking performance [50].

Consider a one-dimensional random process with pdf  $p(x)$ . If  $N_p$  particles  $x^i$ ,  $i = 1, \dots, N_p$  are drawn from the process, then a discrete approximation of the density  $p(x)$  is written as

$$p(x) \approx \sum_{i=1}^{N_p} w^i \delta(x - x^i), \quad (25)$$

where  $\delta(\cdot)$  is the Dirac delta function and  $w^i$  is the weight of the  $i$ -th particle. A particle filter operates by recursively propagating a set of  $N_p$  particles through the model (1)-(2). Under certain assumptions, it has been shown [22] that the discrete density approximation approaches the true *a posteriori* density  $p(\mathbf{x}(k)|\mathbf{z}(1), \dots, \mathbf{z}(k))$  as  $N_p \rightarrow \infty$ . Variations in particle filters differ in the details of the numerical procedures used to recursively propagate the set of particles. Several versions of the algorithms can be found in [2], [20], [22].

Studies of advanced filtering algorithms are numerous and, as a rule, include comparisons between Kalman filter, EKF, UKF, and particle filters, showing tradeoffs among these approaches in different applications. For example, a nonlinear trajectory tracking problem in a one-dimensional state space has been proposed as a benchmark example in [2] where it was shown that classical Kalman or extended Kalman filters do not perform as well as a particle filter. Another example is tracking (from radar observations) a ballistic object on reentry, considered a highly complex problem in nonlinear filtering. In [63] the Cramer-Rao lower bounds were derived for the variance of the estimation error and a comparison of several nonlinear filtering techniques was presented (Kalman filter, EKF, UKF, and the particle filter). Considering the computational and statistical performance, the unscented Kalman filter was shown to be the preferred choice for this particular application.

In many tracking systems and scenarios, targets may switch between a number of maneuvering and non-maneuvering modes and the number and type of sensors

supporting a particular target track can also vary due to the target maneuvering mode or the type and resource limitations of the individual sensors. This variability of the target motions and the sensor systems complicates the tracking process. A Kalman filter is often employed to filter the position measurements for estimating the position, velocity, and acceleration of a target. When designing a Kalman filter, the process (or acceleration) noise  $\mathbf{Q}$  is selected such that the 65 – 95% probability region contains the maximum acceleration level of the target [15]. When targets maneuver, the acceleration changes in a deterministic manner, the white noise assumption associated with the process noise does not hold, and the filter develops a bias in the state estimate during maneuvers. If a “larger”  $\mathbf{Q}$  is chosen, the bias in the state estimates is less during maneuvers, but such a  $\mathbf{Q}$  poorly characterizes the target motion when the target is not maneuvering, leading to filter performance that is far from optimal.

The Interacting Multiple Model (IMM) algorithm can be applied in such scenarios [15], [16]. When the motion of the target is represented by multiple models which are hypothesized to include the correct one, the IMM estimation algorithm is an efficient approach for merging the different model hypotheses. In the IMM algorithm, the state estimate is computed under each possible model hypothesis for the most recent sampling period with each model using a different combination of previous model-conditioned estimates. All the hypotheses are then merged to compute the output state estimate and associated error covariance. The IMM algorithm consists of a filter for each model, a model probability evaluator, an estimate mixer at the input of the filters, and an estimate combiner at the output of the filters. With the assumption that the model switching is governed by an underlying Markov chain, the mixer uses the model probabilities and the model switching probabilities to compute a mixed estimate for each filter.

Define  $\hat{\mathbf{x}}^q(k-1|k-1)$  as the *a posteriori* estimate for model  $q$  at time  $k-1$ . Define  $\omega_q(k)$  as the probability that the target is defined by model  $q$  at time  $k-1$ . Now define a “mixing probability”,  $\omega_{r|q}(k-1|k-1)$  as the probability that the model was defined by model  $r$  at  $k-1$  given that it is now defined by model  $q$ . This can be calculated using Bayes’ rule

$$\omega_{r|q}(k-1|k-1) = \frac{1}{c_q} p_{qr} \omega_r(k-1), \quad q, r = 1, \dots, I_m \quad (26)$$

where  $I_m$  is the number of models,  $p_{qr}$  is the probability of the target switching from model  $r$  to model  $q$  and  $c_q$  is a normalization constant.

Now create the *a priori* state estimate for model  $q$  by using an average of the state estimates from each model, weighted by the mixing probabilities,

$$\hat{\mathbf{x}}^{0q}(k-1|k-1) = \sum_{r=1}^{I_m} \omega_{r|q}(k-1|k-1) \hat{\mathbf{x}}^r(k-1|k-1) \quad (27)$$

$$\begin{aligned} \mathbf{P}^{0q}(k-1|k-1) &= \sum_{r=1}^{I_m} \omega_{r|q}(k-1|k-1) \cdot \\ & (\mathbf{P}^r(k-1|k-1) + \\ & [\hat{\mathbf{x}}^r(k-1|k-1) - \hat{\mathbf{x}}^{0q}(k-1|k-1)] \cdot \\ & [\hat{\mathbf{x}}^r(k-1|k-1) - \hat{\mathbf{x}}^{0q}(k-1|k-1)]') . \end{aligned} \quad (28)$$

Note that these estimates will be different for each model, because the mixing probabilities will be different for each model. Once the *a priori* estimates are calculated, each filter is processed independently using the measurements from the current scan. The model probabilities are updated by calculating the likelihood  $\Delta_q(k)$  that each model produced the current target measurement and combining it with the *a priori* probability of being the model currently that defines the target motion,

$$\omega_q(k) = \frac{1}{c} \sum_{r=1}^{I_m} p_{qr} \omega_r(k-1) . \quad (29)$$

This process is repeated for each scan. To extract an updated state and covariance estimate, the model outputs are merged by averaging the state estimates weighted by their model probabilities:

$$\begin{aligned} \hat{\mathbf{x}}(k|k) &= \sum_{r=1}^{I_m} \omega_r(k) \hat{\mathbf{x}}^r(k|k) \quad (30) \\ \mathbf{P}(k|k) &= \sum_{r=1}^{I_m} \omega_r(k) (\mathbf{P}^r(k|k) + [\hat{\mathbf{x}}^r(k|k) - \hat{\mathbf{x}}(k|k)] \cdot \\ & [\hat{\mathbf{x}}^r(k|k) - \hat{\mathbf{x}}(k|k)]') . \end{aligned} \quad (31)$$

Improvements in the application of the IMM algorithm for radar management (pointing, scheduling, and waveform selection) and highly maneuvering targets in the presence of false alarms and electronic counter measures are presented in [44]. The IMM estimator was used in combination with the probabilistic data association (PDA) technique to compare several benchmark scenarios of multiple sensors tracking a single target. The combined IMM-PDAF technique presents a unified framework for target tracking and radar management (estimation results were used quantitatively to select the operating conditions of the radar). More recently, IMM was applied to improve algorithms for resolving multipath problems (due to reflections) which occur in tracking low elevation targets near the sea-surface [69].

### III. DATA ASSOCIATION

In this section, we overview several approaches for data association. A centralized processing architecture is assumed, where all sensor measurements are received by a central processor prior to data fusion.

Consider tracking  $T$  targets with a single sensor in a cluttered environment. To distinguish between different targets, augment the Kalman filter equations with notation

specifying which target is being referenced [8], [11], [56], [58]:

$$\mathbf{x}^t(k+1) = \mathbf{F}^t(k) \mathbf{x}^t(k) + \mathbf{G}^t(k) \mathbf{w}^t(k) . \quad (32)$$

The noise vectors  $\mathbf{w}^t(k)$  are independent Gaussian random variables with zero mean and known covariances  $\mathbf{Q}^t(k)$ .

Measurements (also called reports or returns) from the sensor are received by a central processor at discrete time intervals. Each measurement can originate from at most one target. Sensor may not provide measurements at every interval. Some of the measurements arise from targets, and some from clutter; some targets may not yield any measurements at all in a particular time interval. The probability of detection is assumed to be constant across targets for a given sensor and will be denoted  $P_D$ . False measurements (clutter) are uniformly distributed throughout the surveillance region with density  $\lambda$ .

Let  $M_k$  be the number of measurements from the sensor at the  $k$ -th time interval. Assuming a pre-correlation gating process is used to eliminate some of the returns [8], [58], let  $m_k$  denote the number of validated measurements from the sensor at time  $k$ . The volume of the gate at time  $k$  is chosen such that with probability  $P_G$  the target-originated measurements, if there are any, fall into the gate of the sensor. The number of gated false measurements is modeled by a Poisson distribution  $\mu_F(m_k)$  given by

$$\mu_F(m_k) = e^{-\lambda V_k} \frac{(\lambda V_k)^{m_k}}{m_k!} . \quad (33)$$

The target-originated position measurements are determined by

$$\mathbf{z}_\ell^t(k) = \mathbf{H}(k) \mathbf{x}^t(k) + \mathbf{v}_\ell^t(k) , \quad (34)$$

where  $1 \leq t \leq T$  and  $1 \leq \ell \leq M_k$ . The  $\mathbf{H}(k)$  matrices are known, each  $\mathbf{v}_\ell^t(k)$  is a zero-mean Gaussian noise vector uncorrelated with all other noise vectors, and the covariance matrices  $\mathbf{R}(k)$  of the noise vectors  $\mathbf{v}^t(k)$  are known. For a given target  $t$  it is not known which measurement  $\ell$  ( $1 \leq \ell \leq M_k$ ) originates from the target. That is the problem of data association whereby it is necessary to determine which measurements originate from which targets [8].

Let  $\mathcal{Z}(k)$  denote the set of gated measurements at time  $k$ ,

$$\mathcal{Z}(k) = (\mathbf{z}_1(k), \dots, \mathbf{z}_{m_k}(k)) . \quad (35)$$

The superscripts  $t$  are not indicated, since it is not known which measurements originated from which target. Finally, let  $\mathcal{Z}^k$  denote the sequence of the first  $k$  observations,

$$\mathcal{Z}^k = (\mathcal{Z}(1), \dots, \mathcal{Z}(k)) . \quad (36)$$

We generally assume that the initial states of the targets are perfectly known, and each target is always well inside the surveillance region. In practical KF-based data association algorithms, gating is usually required to limit computational burden. Assuming that the measurement noise and model process noise are normally distributed, gating is done by a Chi-squared test [8] to determine whether a given measurement lies within the minimum-volume hyper-ellipsoid which contains a set percentage ( $P_G$ ) of the



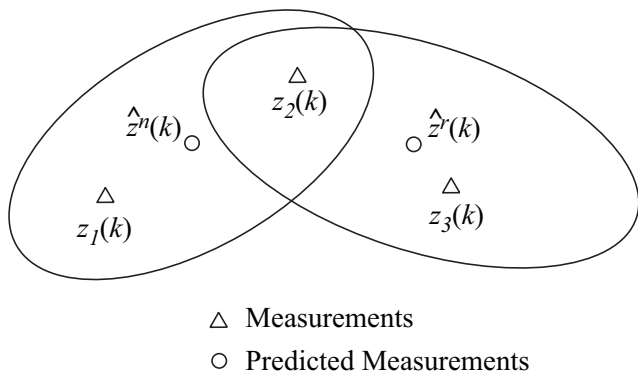


Fig. 3. Tracking multiple targets in the same environment may cause overlapping of validation regions and create sharing of measurements. The gating regions for targets  $n$  and  $r$  are centered around the predicted measurements  $\hat{z}^n(k)$  and  $\hat{z}^r(k)$ . In this example, measurement  $z_2(k)$  is gated by both targets  $n$  and  $r$ .

probability distribution of the predicted measurement. More rigorously, for a given target  $t$ , a gated measurement  $\ell$  must satisfy  $d_\ell^t < \gamma$ , where

$$d_\ell^t(k)^2 = \nu_\ell^t(k) \mathbf{S}^t(k)^{-1} \nu_\ell^t(k) \quad (37)$$

where  $d_\ell^t$  is known as the normalized distance,  $\nu_\ell^t$  is the innovation between target track  $t$  and the  $\ell$ -th measurement during scan  $k$ , and  $\mathbf{S}^t(k)$  is the covariance of the innovation for each measurement. The threshold  $\gamma$  is set such that the true target track will be gated with probability  $P_G$ .

In each time interval where measurements are received, several measurements may be associated with (fall into the gate of) each target as shown in Figure 3. The associated measurements give rise to a Gaussian mixture of components representing several possible estimates for each target state. In the JPDA algorithm, each Gaussian mixture is approximated by a single Gaussian mode with the same mean and covariance as the original mixture [8], [25].

There are several approaches for constructing optimal and sub-optimal estimators in cluttered environments and here we will briefly discuss two categories: Maximum *A Posteriori* (MAP) Probability and Maximum Likelihood (ML) filters. The Probabilistic Data Association (PDA) Filter [8] and Mixture Reduction (MR) filter [65] use a weighted average based on the total probability theorem. The Nearest Neighbor (NN) filter [8] is an approximate MAP filter, whereas the Mean-Field Event-Averaged Maximum Likelihood Estimator (MFEAMLE) [38] is an ML estimator. The Probabilistic Multi-Hypothesis Tracking filter (PMHT) [72] is a hybrid of the two. MFEAMLE and PMHT are inherently multitarget algorithms, but there are multitarget extensions of the NN, PDA, and MR algorithms: the Global Nearest Neighbor filter, Joint Probabilistic Data Association (JPDA) [8], and Multi-Target Mixture Reduction [59].

The challenge, of course is to correctly associate each target's track with the measurements from that target, as opposed to measurements from clutter or another target. This is done by evaluating the probability that each measurement came from a given target, which is based on the

statistical distance between the propagated track's predicted measurement and each measurement received during each scan.

The Nearest Neighbor algorithm [8] selects the measurement that is closest (minimizing  $d_\ell^t(k)^2$ ) to the target track at each scan as the "true" measurement for the Kalman filter. However, when the clutter density is high, or targets are closely-spaced, it is easy to associate the wrong measurement with the target.

To reduce the effects of misassociation, the probabilistic data association (PDA) algorithm [8], [11] and its extension for multiple targets, joint probabilistic data association (JPDA), use a weighted average of gated measurements rather than selecting a single measurement candidate. Given the gated (true and false) measurements, the JPDA tracking algorithm computes the probabilities of each of the measurements being the true measurement of target  $t$ . A combined measurement is then calculated as a weighted average of these gated measurements,

$$\bar{z}^t(k) = \sum_{\ell=1}^M \beta_\ell^t(k) z_\ell(k), \quad (38)$$

where the weights,  $\beta_\ell^t$ , are the computed probability that measurement  $\ell$  originated from target  $t$ . Each combined measurement is then used to update the target state estimate using the Kalman filter. The state covariance resulting from using the combined innovation is

$$\mathbf{P}^t(k|k) = \mathbf{P}^t(k|k-1) - (1 - \beta_0^t) \mathbf{K}^t(k) \mathbf{H}(k) \mathbf{P}^t(k|k-1) + \tilde{\mathbf{P}}^t(k) \quad (39)$$

$$\tilde{\mathbf{P}}^t(k) = \mathbf{K}^t(k) \left[ \sum_{\ell=1}^{m_k} \beta_\ell^t(k) \nu_\ell^t(k) \nu_\ell^t(k)' - \tilde{\nu}^t(k) \tilde{\nu}^t(k)' \right] (\mathbf{K}^t(k))' \quad (40)$$

where  $\beta_0^t$  is the probability that none of the measurements originated from target  $t$ .

In the multisensor case, these algorithms have been extended by typically using recursions in the number of sensors  $N_s$ . As with the filtering algorithms of the previous section, the two alternative implementations of multisensor algorithms are sequential and parallel, *e.g.*, multisensor JPDA (MSJPDA). In practice, sequential algorithms are not only superior in computational requirements over the parallel implementation (in case of MSJPDA, [26], [58]), but in tracking performance as well. It was shown that computational complexity for the parallel implementation (which is not practical in reality due to the fact that multiple sensors are hard to synchronize) grows exponentially with the number of sensors. A sequential implementation of the multisensor JPDA algorithm was presented in [26], where it was shown that it has only linear growth in complexity with an increasing number of sensors, and that it results in better performance (in terms of both root-mean squared position error (RMSE) and track lifetime metrics) for tracking in cluttered environments. While parallel and sequential implementations for pure Kalman filtering (*i.e.*, no clutter

and hence no data association required) are equivalent in terms of performance [81], this is not true for tracking in cluttered environments.

The implicit assumption is that the probability density of the target track can be represented by a single Gaussian distribution or “mode”. The Mixture Reduction Filter relaxes this constraint and propagates multiple distribution modes for each target [65]. Each mode essentially represents a separate target estimate.

Allowing an infinite number of distribution modes, the optimal maximum *a posteriori* estimator is, in the multiple target case, Multiple Hypothesis Testing (MHT). This approach results in an exponential increase in the number of track modes over time [8], as a target track (mode) and an associated probability are created in each time step for all possible combinations of the present set of tracks and the set of new measurements. Associating every new measurement with every target mode is generally impractical for more than a few iterations and hence, various suboptimal filters have been proposed, namely pruning and combining algorithms. Pruning algorithms are characterized by eliminating new modes with lower probabilities in each time step. In combining algorithms, two or more modes are combined into one. In ([5], Ch. 2) an optimization-based approach for the MHT using multidimensional assignment as opposed to the conventional, enumerative approach is introduced.

MFEAMLE, a maximum likelihood algorithm, performs data association using a mean-field approach to sum over all associations [38]. The mean-field approach simplifies the summation over all the associations and avoids assignment of *a priori* probabilities to the associations through the use of a continuous variable,  $\eta$  whose integral converges to the Kronecker delta function. Estimation of the state  $\hat{\mathbf{x}}(k)$  is done by maximizing a pseudo-log likelihood function based on the state and  $\eta$ . However, this function must be solved using (computationally complex) numerical methods.

PMHT is an empirical Bayesian algorithm [72] that includes the simplifying assumption that different target-to-measurement assignments are independent, which is generally not the case, but allows for a reduction in the computational growth of algorithm the relative to those that do not use this assumption (*e.g.*, JPDA). Association of measurements with tracks is modeled as random variables estimated jointly with the target states, and the ML estimate is found iteratively using the method of Expectation Maximization (EM). In practice this is done as a batch process (taking multiple scans of data at once) and replacing the measurement and measurement covariance in the Kalman filter equation with a “synthetic” measurement  $\tilde{\mathbf{z}}^t(k)$  and a corresponding measurement covariance  $\tilde{\mathbf{R}}^t(k)$ . For a set of measurements collected between time  $k_1$  and  $k_2$ ,

$$\tilde{\mathbf{z}}^t(k) = \frac{\sum_{m=1}^{m_k} w_{m,n}^t(k) \mathbf{z}_m(k)}{\sum_{m=1}^{m_k} w_{m,n}^t(k)}, \quad k = k_1, \dots, k_2 \quad (41)$$

$$\tilde{\mathbf{R}}^t(k) = \frac{\mathbf{R}(k)}{\sum_{m=1}^{m_k} w_{m,n}^t(k)} \quad (42)$$

where  $w_{m,n}^t(k)$  is the probability that measurement  $m$

comes from target  $t$  at time  $k$ , and  $n$  is the iteration number. Having calculated the synthetic measurement data, apply a Kalman smoothing algorithm (a smoothing algorithm filters the incoming data moving both forward in backward in time to create a more accurate, “smoothed” state estimate [7]). Repeat the calculation of the synthetic measurements using the new, smoothed state estimates and repeat until some stopping criterion is reached.

In multisensor centralized tracking systems, measurements are typically collected in “scans” or “frames” and then transmitted to a processing center. Communication network delays and varying pre-processing times at the sensor platforms can result in out-of-sequence measurements (OOSMs) [10], [48], [49]. The algorithms that account for OOSMs can be classified in two types, OOSM filtering algorithms and OOSM tracking algorithms. OOSM filtering addresses the update of the state and covariance using the OOSMs. Research in multisensor multitarget OOSM tracking involves data association, filtering, and hypothesis management. The resulting problem (namely, how to update the current state estimate with an “older” measurement) is a nonstandard estimation problem. It was first solved for the case where the out-of-sequence measurement lag is less than a sampling interval (the single-lag case). Subsequently, the suboptimal algorithm was extended to the case of an arbitrary (multistep) lag [10]. A single-model multiple-lag OOSM algorithm for data association, likelihood computation, and hypothesis management for a multisensor multitarget multi-hypothesis tracking (MHT) system that handles missed detections and clutter is presented in [49]. More recently, particle filters have also been applied to OOSM problems [50].

#### IV. SENSOR REGISTRATION

An added complication to the fusion of multiple sensor outputs is the effect of relative errors on the target tracks. Most sensors measure target location relative to their own position (for example a radar might infer the distance from the radar antenna to the target from the time-of-flight of an electromagnetic pulse). In many applications, only the relative distance from the platform to the target is important. In those cases, the target position can be represented on a coordinate system centered on the sensor. However, if the location of the target relative to some other location is important, then a more general, global coordinate system is required (as when an airborne radar sentry relays a target’s location to a ground-based or sea-based tracking system). In this case, the location of the sensor in the global coordinate system becomes as important as that of the target. Any error in the estimate of the sensor’s position will become an error in the estimate of the target’s position. If multiple targets are tracked by that sensor, they will all have a correlated error that is a function of the sensor position error. When fusing the outputs of multiple sensors, it will generally be the case that these sensors are not all located in one position. Thus a global coordinate system is required, as well as accurate locations of each sensor. The process of eliminating such errors is known as sensor registration.



In its most general sense, sensor registration is another state estimation problem, only in this case the state being estimated is that of the sensor platform, regardless of whether it is stationary or mobile. The most straightforward approach is to measure the location of a target whose position is very well known, and use that measurement to correct the estimate of the sensor's location. Of course for mobile platforms this will involve repeated measurements of the reference target as well as some sort of predictive filter such as the Kalman filter to maintain an accurate estimate.

One way to accomplish registration for both stationary and moving platforms is by augmenting the target's state with that of the sensor platform's (*e.g.*, [53], [64]). Thus the same tracking filter that is being used to locate the target will do the same for the sensor platform.

The addition of multiple targets tracked by each sensor increases the complexity of the sensor registration task. Now the different target tracks must be correlated with one another to determine which tracks correspond to the same objects. However, if the tracks can be correctly associated, the relative bias of the different sensors can be identified and eliminated — relative registration. Several methods of track correlation and bias estimation exist, including [46], [47], [71]. The issues of sensor bias observability and absolute registration with only targets of opportunity are discussed in [46], [47].

## V. DISTRIBUTED VS. CENTRALIZED TRACKING

While the fusion of measurements from multiple sensors can dramatically improve tracking accuracy, there is a cost in computational demand as well as communication bandwidth requirements. Each measurement must be transmitted to a central processor to be processed, representing a significant burden in high clutter or target-rich environments. An alternative architecture is shown in Figure 4, where a single processor has been replaced by a group of processors, each of which monitors a subset of the total number of sensors and reports the results to a single, global processor. By distributing the tracking tasks among a group of processors, the computational demand on any one processor can be reduced. Furthermore, if those processors can send only the target tracks, rather than each measurement, then the communication requirements would also be dramatically reduced (even more so if track information can be communicated at a reduced rate). Additionally, such an architecture will result in a more robust tracking system as the failure of a single processor no longer represents a catastrophic failure.

The drawback to this approach is that because each processed track includes the effects of process noise originating with the target itself, the errors among the tracks are now correlated [4]. There are several approaches to distributed track fusion, including both maximum-likelihood (*e.g.*, [6]) and minimum mean square error (MMSE) solutions. Here, we will overview two approaches to MMSE track fusion.

The first approach, direct track-to-track fusion, includes the correlation effects directly in the fusion process. Applying this approach to the linear case, with two sensor-level

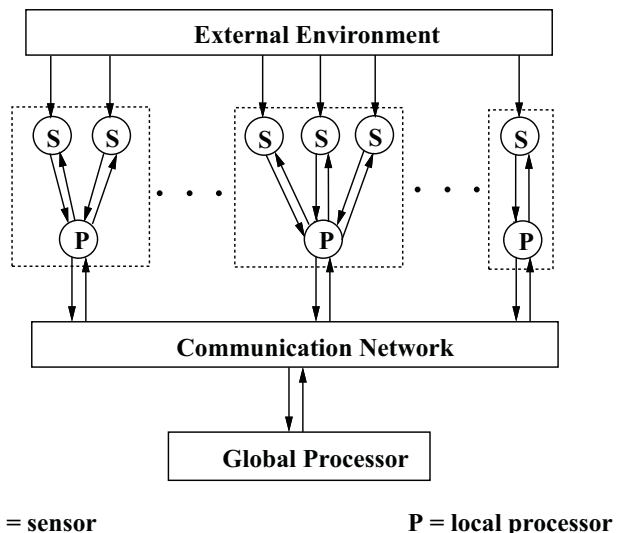


Fig. 4. A distributed architecture divides the tracking task among multiple processors to reduce communication burden and the computational demand on individual processors.

tracks  $i$  and  $j$  being fused at the global level, the state estimate becomes [19]

$$\begin{aligned} \hat{\mathbf{x}}(k|k) = & \mathbf{P}(k|k) (\mathbf{P}_i^{-1}(k|k)\hat{\mathbf{x}}_i(k|k) \\ & + \mathbf{P}_j^{-1}(k|k)\hat{\mathbf{x}}_j(k|k) \\ & - \mathbf{P}_i^{-1}(k|k-1)\hat{\mathbf{x}}_i(k|k-1) \\ & - \mathbf{P}_j^{-1}(k|k-1)\hat{\mathbf{x}}_j(k|k-1) \\ & + \mathbf{P}^{-1}(k|k-1)\hat{\mathbf{x}}(k|k-1)) \end{aligned} \quad (43)$$

$$\begin{aligned} \mathbf{P}(k|k) = & (\mathbf{P}_i^{-1}(k|k) + \mathbf{P}_j^{-1}(k|k) - \mathbf{P}_i^{-1}(k|k-1) \\ & - \mathbf{P}_j^{-1}(k|k-1) + \mathbf{P}^{-1}(k|k-1))^{-1} \end{aligned} \quad (44)$$

where  $\hat{\mathbf{x}}_i(k|k)$ ,  $\hat{\mathbf{x}}_j(k|k)$ , and  $\hat{\mathbf{x}}(k|k)$  are the updated state estimates of the tracks from processor/sensor  $i$ ,  $j$ , and the global processor, respectively, while  $\hat{\mathbf{x}}_i(k|k-1)$ ,  $\hat{\mathbf{x}}_j(k|k-1)$  and  $\hat{\mathbf{x}}(k|k-1)$ , are the *a priori* predictions of the state from the previous scan, and  $\mathbf{P}_i(k|\cdot)$ ,  $\mathbf{P}_j(k|\cdot)$ , and  $\mathbf{P}(k|\cdot)$  are the respective covariances. This approach is only exact when the updates from the processors are synchronized and take place at every scan. If the updates from the sensor-level processors are not passed to the global processor every scan, then the propagation of the common process noise again causes correlation between the estimation errors. This track-to-track fusion approach has not yet been extended to account for clutter and data association as well as more general classes of architectures.

An alternative approach<sup>1</sup> is known as decorrelation, where the correlation between sensor-level tracks is removed, allowing the global processor to treat the incoming target tracks as uncorrelated measurements. Although originally derived for pure filtering in [23], [27], it has recently

<sup>1</sup>This is really an algebraic rearrangement of (43)–(44); see Eq. (8.6.1-9) in [8].

been extended to address tracking targets in clutter [40]. In this approach, decorrelation is achieved by subtracting the *a priori* component of each state estimate

$$\mathbf{y}(k) = \hat{\mathbf{x}}(k|k) - \mathbf{C}(k)\hat{\mathbf{x}}(k|k-1) \quad (45)$$

$$\mathbf{C}(k) = \mathbf{P}(k|k)\mathbf{P}^{-1}(k|k-1) \quad (46)$$

$$\mathbf{Y}(k) = \mathbf{P}(k|k) - \mathbf{P}(k|k)\mathbf{P}^{-1}(k|k-1)\mathbf{P}(k|k) \quad (47)$$

where  $\hat{\mathbf{x}}(k|k-1)$  is the *a priori* target state estimate,  $\mathbf{P}(k|k-1)$  is the *a priori* covariance of that state estimate,  $\mathbf{y}(k)$  is the decorrelated state estimate, and  $\mathbf{Y}(k)$  is the covariance of that estimate. As a ‘‘pseudomeasurement’’,  $\mathbf{y}(k)$ , along with its covariance, are sent from each local processor for each track to a global processor. The measurement matrix for the decorrelated estimate becomes

$$\mathbf{B}(k) = \mathbf{I} - \mathbf{C}(k) \quad (48)$$

$$= \mathbf{I} - \mathbf{P}(k|k)\mathbf{P}^{-1}(k|k-1) \quad (49)$$

where  $\mathbf{I}$  is the identity matrix. Because the  $\mathbf{y}(k)$  sequences are decorrelated, the global processor can use well-established ‘centralized’ fusion algorithms for processing the  $\mathbf{y}(k)$  to form global estimates of the target states. This technique has been further extended for multisensor multitarget tracking of targets in clutter [40], [41], including feeding back information from global to local processors [42].

## VI. SENSOR MANAGEMENT

Sensor managers are a general class of systems that generate sensing actions, then prioritize and schedule those actions [52]. Sensing actions can include the tasking of sensors to illuminate a target, the selection of sensor modes, or scanning an area for unknown targets. These actions will be selected to achieve various goals such as maintaining a target track, optimizing the chance of detecting new targets, identifying detected targets, and minimizing electromagnetic emissions to reduce the chance of detection by the enemy. Since sensing resources are generally limited, sensor managers must solve an optimization problem that balances those resources with tracking goals. A discussion of the various sensor manager approaches becomes essentially a discussion on how to formulate the optimization problem. Although much work has been done on the management of individual sensors, especially radar (where the sampling or revisit rate [44], [13], [21], [43], [79], [80], or emitted waveforms [39] can be modified to control the level of uncertainty in the target track), since this tutorial focuses on multisensor fusion, only the selection of sensors from a larger group of sensors will be discussed.

Figure 5 shows how a sensor manager is incorporated into a target tracking system. The target state estimate covariance is used to select the appropriate set of sensors to use on a given target. Those sensors then produce measurements that will be used to update the state estimate of that target. The sensors are selected based on their ability to optimize some function of either the resulting target estimate covariances

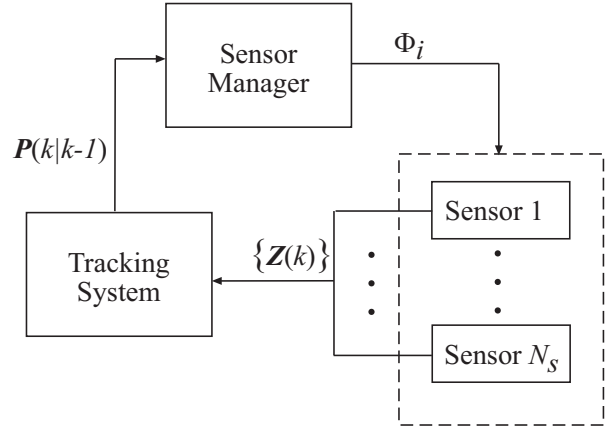


Fig. 5. Block diagram of a multisensor tracking system with a sensor manager. The *a priori* covariance from the target tracker is passed to the sensor manager which then selects a subset  $\Phi_i$  of the sensor suite to use during the following scans. Sensors can be chosen to maximize the information gain from each sensor [66], achieve a desired track covariance [36], or to optimize some other tracking metric.

across all targets or the amount of information from the sensors. This approach can generally be defined by

$$\Phi_i = \arg \min_i f(\mathbf{P}(k|k-1), \mathbf{P}_i(\cdot|k)) \quad (50)$$

where  $\Phi_i$  is sensor combination  $i$ ,  $\mathbf{P}(k|k-1)$  is the *a priori* covariance of the target track, and  $\mathbf{P}_i(\cdot|k)$  is the *a posteriori* covariance (possibly propagated forward in time) after the application of sensor combination  $i$ . The possible cost functions include one consisting of the trace of each target’s state estimate covariance weighted by target priority (which presumably increases with importance) [54],

$$f(\mathbf{P}(k|k-1), \mathbf{P}_i(k+1|k)) = \text{priority} \times \text{Tr}(\mathbf{P}_i(k+1|k)) \quad (51)$$

as well as one that uses a spectral decomposition of the covariance matrix to select a sensor that will reduce the largest eigenvalue of that matrix [57].

The 1990’s brought the application of information theory to the sensor management field, which has dominated it ever since. Here, instead of minimizing a cost function as in (51), the goal is to maximize the information gained by each sensor. Information gain is defined by taking the natural log of the quotient formed by dividing the determinant of the *a priori* covariance by the determinant of the *a posteriori* covariance for each sensor combination [66]:

$$I(\mathbf{P}(k|k-1), \mathbf{P}_i(k|k)) = \frac{1}{2} \ln \frac{|\mathbf{P}(k|k-1)|}{|\mathbf{P}_i(k|k)|} \quad (52)$$

This amounts to a measure of the uncertainty volume reduction. Other, similar definitions of information gain have also been proposed [31], [68]. The use of the information metric has been extended to include multitarget tracking scenarios requiring data association [67].

The information metric is particularly useful in that it can be applied to a range of sensing activities, allowing the sensor manager to compare the utility of a variety of activities under one metric. For example, information resulting from a target search can be calculated based on the difference in the probability of a given cell containing a target depending on whether the sensor will or will not detect a target there. When combined with the information metric of [66], the tasks of target search and tracking can be combined into a single sensor manager [51].

A relatively new approach to sensor management is known as Covariance Control. This approach defines a desired target estimate covariance,  $\mathbf{P}_d$ , for each track and attempts to achieve that covariance. The desired covariance can be a function of the target's priority or of an external goal such as achieving enough estimate accuracy before applying weapons to that target. A variety of approaches based on these metrics exist [36]. One metric, known as the Eigenvalue/Minimum Sensors Algorithm, attempts to achieve the desired covariance goal while minimizing the number of sensors used. The fact that achieving  $\mathbf{P}_d$  will result in the difference  $\mathbf{P}_d - \mathbf{P}_i$  having all positive eigenvalues, gives this algorithm its name. More rigorously, the algorithm selects a sensor combination  $i_{ev}$  with the fewest sensors that achieves the covariance goal:

$$\Phi_{ev} = \{\Phi_i : \mathbf{P}_d - \mathbf{P}_i > 0\} \quad (53)$$

$$i_{ev} = \arg \min_i |\Phi_i|, \quad \Phi_i \in \Phi_{ev} \quad (54)$$

where  $|\Phi_i| = N_{s_i}$  is the number of sensors in combination  $i$ . This approach has been extended to include the influence of the PDAF algorithm on the covariance [34]. Additionally, algorithms for selecting a desired covariance and the appropriate sensors have been derived to reduce the effect of closely-spaced targets on probabilistic data association methods [35].

## VII. COMPUTATIONAL COMPLEXITY AND REAL-TIME IMPLEMENTATION ISSUES

The view of the computational complexity for different tracking algorithms is affected by their intended application. An understanding of the tradeoffs between the performance and computational complexity of the applied algorithms may provide guidance for users to select the best algorithms for their purposes.

One such study presents the computational requirements and the performance of a parallel implementation of the multisensor JPDA algorithm [26], [58]. A sequential implementation of the multisensor JPDA algorithm was presented in [26], and it was shown that it has linear growth in complexity with an increasing number of sensors, and that it results in better performance (in terms of both root-mean squared position error (RMSE) and track lifetime metrics) for tracking in cluttered environments.

In [60], five data association algorithms were compared in simulation using a common linear single-target motion test bed. The comparison included the NN, PMHT, PDA, MR, and MFEAMLE algorithms described in Section III.

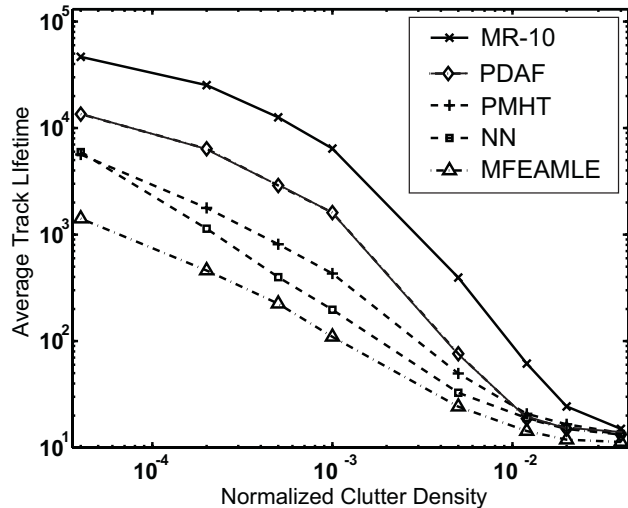


Fig. 6. Average track lifetime as a function of normalized clutter density

The performance measures were: track lifetime (Figure 6), root-mean squared position error (RMSE) (Figure 7), and computational complexity (Figure 8), while the variable parameters were normalized target acceleration and normalized clutter density. Figures 6, 7, and 8 present results for tracking a single target until it is lost.

The definition of track loss used is that proposed in [65]: a track is determined to be lost when either the estimate is sufficiently far from the true track (the “norm test”) or the true measurement falls outside of the gate (the “acceptance test”) for five consecutive time steps. Track lifetime is the difference between the time step of track loss and the time step of track initiation (assumed to be at time zero, since track initiation was not considered in [60]). Track lifetimes for a given normalized clutter density were averaged across all trials.

The reported results show some expected trends, *e.g.*, the nearest neighbor algorithm has acceptable performance for low and medium clutter, has low computational burden, but tracks poorly in high clutter; MR with 10 modes performs well in terms of track lifetime, but at the cost of higher computational complexity compared to other algorithms, whereas the PDA algorithm yields a “middle of the road” solution; PMHT shows superior performance in the RMSE metric at the cost of additional computational burden across the clutter range; and the MFEAMLE algorithm shows high computational demand due to inherent numerical issues.

The rise in computational complexity with the number of sensors is an issue for sensor managers as well. A global search over all possible sensor combinations grows exponentially with the number of sensors, often outstripping the computational cost of simply using all of the sensors (other limitations, such as a maximum number of sensor dwells, may make sensor management necessary even if the computational demand is large). There are however, several search methods that can reduce this growth. The most straightforward is the “Greedy” algorithm, which selects

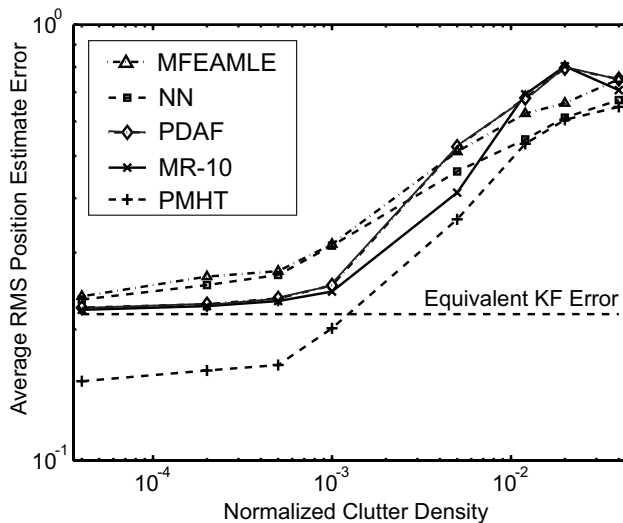


Fig. 7. Average RMS position estimate error as a function of normalized clutter density

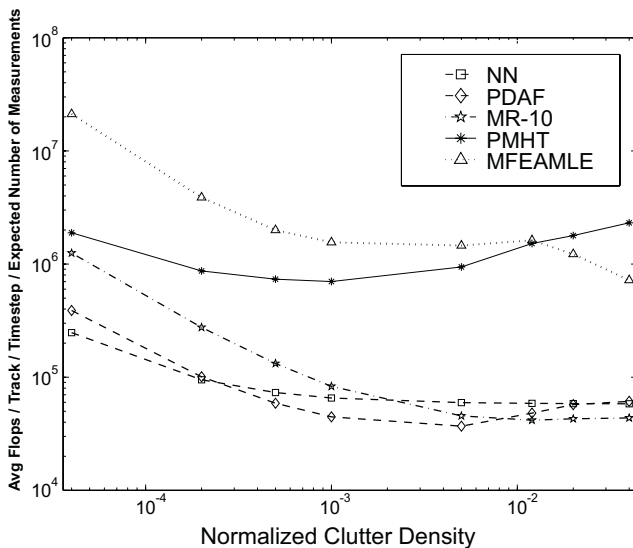


Fig. 8. Computational burden as a function of normalized clutter density

sensors that maximize the change in the cost function, driving the farthest towards its maximum or minimum, depending on the formulation, which reduces the computational demand to a polynomial growth with the number of sensors [36], [82]. Other search techniques add a random component to the search to further improve the performance of the sensor selection algorithm [37].

Comparison of different tracking algorithms or parameter sets within one algorithm relies on time-consuming and computationally demanding Monte Carlo simulations. In [76] a method is presented to minimize simulation time, yet to achieve a desirable confidence of recognizing better designs. It is achieved by applying ordinal optimization ideas (*i.e.*, determining rank order instead of quantifying the performance of the best design) and computing budget allocation techniques [73] (*i.e.*, allocating computational

resources so that better designs get more simulation time). The developed method is applied to a general tracking problem of  $N_s$  sensors tracking  $T$  targets using a sequential multi-sensor data fusion tracking algorithm (MSJPDA). The optimization consists of finding the order of processing sensor information that results in the smallest variance of the position error. Results obtained with high confidence levels and in reduced simulation times show that processing the best available sensor the last in sequential MSJPDA has the smallest RMS error, *e.g.*, the order of processing sensor information affects tracking precision in the presence of clutter (*i.e.*, when data association algorithm is necessary). The presented method can be applied to any ranking and selection problem where variance (or standard deviation) is the performance metric. Further improvement in comparison of tracking algorithm was achieved when the distribution of position error was modeled with Gaussian mixture pdf [74], [75]. Incorporating the improved statistical modeling of the position error variance in computing budget allocation algorithm [74], simulation time is reduced by an order of magnitude compared to long Monte Carlo experiment.

One application of the developed variance ranking algorithm is comparing sensor-order designs in sequential MSJPDA as a function of clutter density  $\lambda$  and RMSE as performance metric. A design is an ordered set of sensor noise covariance parameters,  $\theta_j = \{r_1, r_2, \dots, r_{N_s}\}$ , with sensor noise covariance  $\mathbf{R}_s = r_s \mathbf{I}$  in (8). A better sensor is characterized with smaller sensor noise parameter  $r$ . In [75], the sequential MSJPDA algorithm with  $N_s = 3$  sensors tracking  $T = 4$  targets are considered. The compared designs are defined as ordered sets of sensor noise parameters (two identical sensors with  $r = 0.1$  and one better sensor with  $r = 0.01$ ). The objective is to find the best design in terms of the RMSE for various clutter densities  $\lambda$ . The three designs are

$$\begin{aligned} \theta_1 &= \{0.1, 0.1, 0.01\}, \quad \theta_2 = \{0.1, 0.01, 0.1\}, \\ \theta_3 &= \{0.01, 0.1, 0.1\}. \end{aligned} \quad (55)$$

The variance ranking tools [73], [75] were used in the optimization algorithm to yield the results in Fig. 9 and Table I. For low clutter densities  $\lambda$ , the order of sensor processing does not affect the tracking performance. Since the RMSE's do not differ significantly, the best ranking design  $\theta_b$  is selected with a small confidence level when the maximum number of simulation runs  $N_{max} = 100$  is exhausted. For higher clutter  $\lambda$ , the order of sensor processing affects the tracking more significantly. The best ranking design is consistently the one where the best sensor is processed last.

## VIII. APPLICATIONS

Multisensor fusion was developed to combine sensor information and to improve target state estimates (over using a single sensor) in a surveillance region, or in other words, to resolve uncertainty in diverse data sets retrieved from



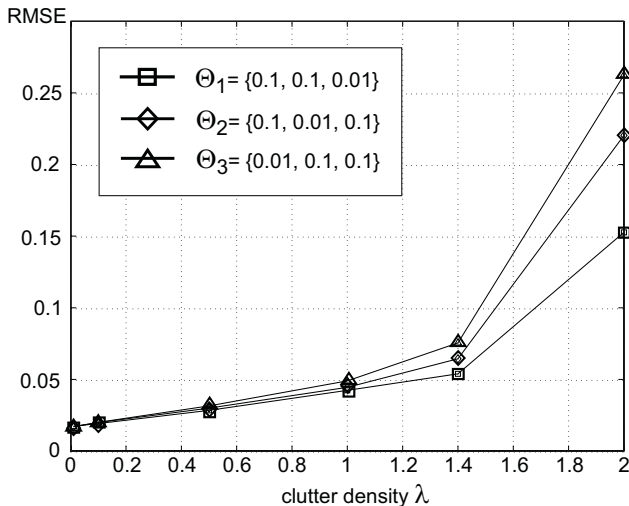


Fig. 9. Root-mean squared position error (RMSE) of the sequential MSJFDA algorithm as a function of clutter density  $\lambda$  for three designs.

$\lambda$	0.01	0.1	0.5	1.0	1.4	2.0
$\theta_b$	$\theta_2$	$\theta_1$	$\theta_1$	$\theta_1$	$\theta_1$	$\theta_1$
$N_{runs}$	100	100	93	85	21	15
conf. level	0.43	0.86	0.98	0.98	0.98	0.99

TABLE I

THE SELECTED BEST RANKING DESIGN  $\theta_b$ , THE TOTAL NUMBER OF SIMULATION RUNS  $N_{runs}$ , AND THE CONFIDENCE LEVEL AS FUNCTIONS OF CLUTTER DENSITY  $\lambda$ . THE MAXIMUM ALLOWED NUMBER OF RUNS IS  $N_{max} = 100$ .

different sensors. Early applications were mostly military, such as automatic target recognition, battlefield assessment, and smart weapons. However, techniques originally developed in multisensor fusion have now found their way in applications and projects of interdisciplinary nature for civilian use (space technology [12], [18] and earth sciences [61]) where some form of uncertainty is built in the acquired data sets.

Sensor fusion algorithms have also been validated with biomedical data [17]. Though the methods for sensor fusion in biomedical applications originate in information processing, artificial intelligence, and classification algorithms, the applied estimation methods combine statistical and control theory algorithms (*e.g.*, weighted least squares, maximum *a posteriori* probability, minimum *a posteriori* variance estimation, etc.). Estimates are recursively updated via Kalman or adaptive Kalman filters. Clinical testing of software for detecting respiratory phases has shown that the complex techniques of sensor fusion have become accepted by physicians to aide diagnostics only after the introduction of user-friendly interfaces. Another step forward has been the implementation of soft computing on a symbol level through which the user can integrate his or her own experience and terminology. More complex medical applications of multisensor fusion include improvement of image processing and pattern recognition, and hence fall in the information fusion category, which is beyond the scope

of this paper.

#### A. Tracking of Ground Targets

The tracking of aerial or sea-based targets usually rely on the assumptions that the motion of each target is constrained only by the motion models, which may include types of motion as well as limits to acceleration and velocity; that the target will generally not be obscured (may not hold in optical tracking); and that target clutter will be uniformly distributed. These assumptions are far less likely to hold when tracking ground targets (see *e.g.*, [24] and [70], p. 423). In these applications, the motion of the target may be constrained by terrain, targets may often be obscured by foliage or other terrain, and clutter distribution is more likely to deviate from uniform. These difficulties may hold for air- and sea-based tracking as well when targets are flying close to the earth or are in littoral areas. One approach for tracking ground targets on-road and off-road has been presented in [45].

There are several ways of dealing with the increased tracking difficulties described above. One is to use moving target indicators or doppler radar to distinguish moving targets from stationary objects ([70], p. 104). While this will reduce the amount of clutter, it does not alleviate obscuration (in this case, a stopped vehicle can be considered obscured) or constraints on motion. Obscuration by foliage can be reduced through the use of lower frequency radars that can see through the tree canopy. The inclusion of terrain information can be used in conjunction with sensor measurements to improve the target state estimates [1], [24], [30]. Roadway intersections lead naturally to the use of MHT or IMM trackers, as well.

Investigation efforts in the development of high-accuracy multisensor vehicle state estimation schemes using the UKF is presented in [77]. The model is able to cope with vehicle slip using multisensor data from internal sensors, odometry, and Differential Global Positioning System (D-GPS).

## IX. SUMMARY AND AREAS OF FUTURE WORK

Multisensor data fusion seeks to combine information from multiple sensors and sources to improve inferences about a surveillance region that are not achievable from a single sensor or source. Applications include military applications such as automatic target recognition, automated situation assessment, and for smart weapons, as well as civilian applications such as automated plant management, mobile robot navigation, and environmental modeling. Thus, an extensive legacy exists including: process models, algorithms, evolving tool kits, and systems engineering methodology (system design and algorithm selection).

There are still a number of growing challenges for data fusion, at lower levels of inferences: understanding sensor processing and sensor limitations, multitarget tracking and identification, and at higher levels of inferences: automated reasoning for situation assessment, development of alternative hypotheses for threat assessment, monitoring, and control of fusion processes.

One potential area of future improvement in sensor fusion algorithms is incorporating the continuously emerging advanced filtering techniques developed to handle the nonlinear nature of tracking while reducing computational burden. Another area is resolving issues in data fusion over distributed sensor networks with data communication delays [55].

## REFERENCES

- [1] D. Andrisani, M. F. Tenorio, J. Lu, and F. P. Kuhl, "A Hierarchical Target Tracker Using Image Data," *Proc. American Control Conf.*, San Diego, CA, pp. 1004-1008, June 1990.
- [2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Trans. Sig. Proc.*, Vol. 50, No. 2, pp. 174-188, Feb. 2002.
- [3] D. Avitzour, "A Stochastic Simulation Bayesian Approach to Multitarget Tracking," *IEE Proc. on Radar, Sonar and Navigation*, Vol. 142, 1995.
- [4] Y. Bar-Shalom, "On the Track-to-Track Correlation Problem," *IEEE Trans. Automatic Control*, Vol. 26, No. 2, pp. 571-572, Apr. 1981.
- [5] Y. Bar-Shalom and W. Dale Blair (editors), *Multitarget-Multisensor Tracking: Applications and Advances*, Vol. III, Artech House, 2000.
- [6] Y. Bar-Shalom and L. Campo, "The Effect of the Common Process Noise on the Two-Sensor Fused-Track Covariance," *IEEE Trans. Aerospace and Electronic Systems*, Vol. 22, No. 6, pp. 803-805, Nov. 1986.
- [7] Y. Bar-Shalom and X. Li, *Estimation and Tracking: Principles, Techniques, and Software*, Artech House, Boston, 1993.
- [8] Y. Bar-Shalom and X. R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, YBS Publishing, 1995.
- [9] Y. Bar-Shalom, X. R. Li and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction*, J. Wiley and Sons, 2001.
- [10] Y. Bar-Shalom, M. Mallick, H. Chen, and R. Washburn, "One-Step Solution for the General Out-of-Sequence-Measurement Problem in Tracking," *Proc. IEEE Aerospace Conf.*, Vol. 4, pp. 1551-1559, March 2002.
- [11] Y. Bar-Shalom and E. Tse, "Tracking in a Cluttered Environment with Probabilistic Data Association," *Automatica*, Vol. 11, No. 5, pp. 451-460, Sept. 1975.
- [12] E. T. Baumgartner and S. B. Skaar, "An Autonomous Vision-Based Mobile Robot," *IEEE Trans. Automatic Control*, Vol. 39, No. 3, pp. 493-502, March 1994.
- [13] S. S. Blackman, M. T. Busch, and R. F. Popoli, "IMM/MHT Tracking and Data Association for Benchmark Tracking Problem," *Proc. American Control Conf.*, Seattle, WA, pp. 2606-2610, 1995.
- [14] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 1999.
- [15] W. D. Blair and Y. Bar-Shalom, "Tracking Maneuvering Targets With Multiple Sensors: Does More Data Always Mean Better Estimates?" *IEEE Trans. Aerospace and Electronic Systems*, Vol. 32, No. 1, pp. 450-456, Jan. 1996.
- [16] H. A. P. Blom and Y. Bar-Shalom, "The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients," *IEEE Trans. Automatic Control*, Vol. 33, No. 8, pp. 780-783, Aug. 1988.
- [17] B. R. Bracio, W. Horn, and D. P. F. Moller, "Sensor Fusion in Biomedical Systems," *Proc. 19th Annual Intern. Conf. of the IEEE Society of Eng. in Medicine and Biology society*, Vol. 3, pp. 1387-1390, 1997.
- [18] S. Brunke and M. Campbell, "Estimation architecture for future autonomous vehicles," *Proc. American Control Conf.*, pp. 1108-1114, May 2002.
- [19] K. C. Chang, R. K. Saha, Y. Bar-Shalom, "On Optimal Track-to-Track-Fusion," *IEEE Trans. Aerospace and Electronic Systems*, Vol. 33, No. 4, Oct. 1997.
- [20] D. Crisan and A. Doucet, "A Survey of Convergence Results on Particle Filtering Methods for Practitioners," *IEEE Trans. Signal Processing*, Vol. 50, No. 3, pp. 736-746, March 2002.
- [21] E. Daeipour, Y. Bar-Shalom, and X. Li, "Adaptive Beam Pointing Control of a Phased Array Radar Using an IMM Estimator," *Proc. American Control Conf.*, Baltimore, MD, pp. 2093-2097, 1994.
- [22] A. Doucet, N. de Freitas, and N. Gordon (editors), *Sequential Monte Carlo Methods in Practice*, Springer-Verlag Inc., 2001.
- [23] O. E. Drummond, "Feedback in Track Fusion Without Process Noise," *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, 2561, pp. 369-383, 1995.
- [24] R. H. Enders, "Fundamentals of on-road tracking," *Acquisition, Tracking, and Pointing*, Orlando, FL, pp. 334-342, 1999.
- [25] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association," *IEEE J. Oceanic Engineering*, Vol. 8, No. 3, pp. 173-183, July 1983.
- [26] C. W. Frei and L. Y. Pao, "Alternatives to Monte-Carlo Simulation Evaluations of Two Multi-sensor Fusion Algorithms," *Automatica*, Vol. 34, No. 1, pp. 103-110, Jan. 1998.
- [27] G. Frenkel, "Multisensor Tracking of Ballistic Targets," *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, 2561, pp. 337-346, 1995.
- [28] L. Frenkel and M. Feder, "Recursive Estimate-Maximize (EM) Algorithms for Time Varying Parameters with Applications to Multiple Target Tracking," *Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP-95*, Vol. 3, pp. 2068-2071, 1995.
- [29] D. L. Hall and J. Llinas, "An Introduction to Multisensor Data Fusion," *Proc. IEEE*, Vol. 85, No. 1, pp. 6-23, 1997.
- [30] J. G. Herrero, J. A. Besada Portas, and J. R. Casar Corredera, "Use of Map Information for Tracking Targets on Airport Surface," *IEEE Trans. Aerospace and Electronic Systems*, Vol. 39, No. 2, pp. 675-693, April 2003.
- [31] K. J. Hintz and E. S. McVey, "Multi-Process Constrained Estimation," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 21, pp. 237-244, 1991.
- [32] S. J. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," *Proc. AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulation and Control*, 1997.
- [33] S. J. Julier, J. K. Uhlmann, and H. D. Durrant-Whyte, "A new approach for filtering nonlinear systems," *Proc. American Control Conf.*, pp. 1628-1632, June 1995.
- [34] M. Kalandros and L. Y. Pao, "The Effects of Data Association on Sensor Manager Systems," *Proc. AIAA Guidance, Navigation, and Control Conf.*, Denver, CO, 2000.
- [35] M. Kalandros and L. Y. Pao, "Sensor Management for Tracking Interacting Targets," *Proc. Workshop on Estimation, Tracking, Fusion, and Control: A Tribute to Yaakov Bar-Shalom*, Monterey, CA, pp. 221-248, May 2001.
- [36] M. Kalandros and L. Y. Pao, "Covariance Control for Multisensor Systems," *IEEE Trans. Aerospace and Electronic Systems*, Vol. 38, No. 4, pp. 1138-1157, Oct. 2002.
- [37] M. Kalandros, L. Y. Pao, and Y. C. Ho, "Randomization and Super-Heuristics in Choosing Sensor Sets for Target Tracking Applications," *Proc. IEEE Conf. on Decision and Control*, Phoenix, AZ, pp. 1803-1808, 1999.
- [38] K. Kastella, "Event-Averaged Maximum Likelihood Estimation and Mean-Field Theory in Multitarget Tracking," *IEEE Trans. Automatic Control*, Vol. 40, No. 6, pp. 1070-1074, 1995.
- [39] D. J. Kershaw and R. J. Evans, "Optimal Waveform Selection for Tracking Systems," *IEEE Trans. Information Theory*, vol. 40, pp. 1536-1550, 1994.
- [40] W. Khawsuk and L. Y. Pao, "Decorrelated State Estimation for Distributed Tracking of Interacting Targets in Cluttered Environments," *Proc. American Control Conf.*, Anchorage, AK, pp. 899-904, May 2002.
- [41] W. Khawsuk and L. Y. Pao, "Decorrelated State Estimation for Distributed Tracking Using Multiple Sensors in Cluttered Environments," *Proc. American Control Conf.*, Denver, CO, pp. 3208-3214, June 2003.
- [42] W. Khawsuk and L. Y. Pao, "Distributed Multi-sensor Multitarget with Feedback," *Proc. American Control Conf.*, Boston, MA, June 2004.
- [43] T. Kirubarajan, Y. Bar-Shalom, and E. Daeipour, "Adaptive Beam Pointing Control of a Phased Array Radar in the Presence of ECM and False Alarms Using IMM/PDAF," *Proc. American Control Conf.*, Seattle, WA, pp. 2616-2620, 1995.
- [44] T. Kirubarajan, Y. Bar-Shalom, W. D. Blair, and G. A. Watson, "IMM/PDAF for Radar Management and Tracking Benchmark with ECM," *IEEE Trans. Aerospace and Electronic Systems*, Vol. 34, No. 4, pp. 1115-1134, 1998.
- [45] T. Kirubarajan, Y. Bar-Shalom, K.R. Pattipati and I. Kadar, "Ground Target Tracking with Topography-Based Variable Structure IMM



- Estimator," *IEEE Trans. Aerospace and Electronic Systems*, Vol. 36, No. 1, pp. 26-46, Jan. 2000.
- [46] X. D. Lin, Y. Bar-Shalom and T. Kirubarajan, "Exact Multisensor Dynamic Bias Estimation with Local Tracks," *Proc. 6th Int. Conf. Information Fusion*, Cairns, Australia, July 2003.
- [47] X. D. Lin, Y. Bar-Shalom and T. Kirubarajan, "Multisensor Bias Estimation Using Local Tracks without A Priori Association", *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, San Diego, CA, August 2003.
- [48] M. Mallick, S. Coraluppi, and C. Carthel, "Advances in Asynchronous and Decentralized Estimation," *Proc. IEEE Aerospace Conf.*, pp. 1873-1888, March 2001.
- [49] M. Mallick, J. Krant, and Y. Bar-Shalom, "Multi-Sensor Multi-Target Tracking Using Out-of-Sequence Measurements," *Proc. 5th Int. Conf. Information Fusion*, pp. 135-142, July 2002.
- [50] M. Mallick, S. Maskell, T. Kirubarajan, and N. Gordon, "Littoral Tracking Using Particle Filter," *Proc. 5th Int. Conf. Information Fusion*, pp. 935-942, July 2002.
- [51] G. A. McIntyre and K. J. Hintz, "An Information Theoretic Approach to Sensor Scheduling," *Proc. SPIE*, Vol. 2755, pp. 304-312, 1996.
- [52] S. Musick and R. Malhotra, "Chasing the Elusive Sensor Manager," *Proc. IEEE 1994 NAECON*, Vol. 1, pp. 606-613, Dayton, OH, IEEE: New York, NY, 1994.
- [53] N. Nabaa and R. H. Bishop, "Solution to a Multisensor Tracking Problem with Sensor Registration Errors," *IEEE Trans. Aerospace and Electronic Systems*, Vol. 35, No. 1, pp. 354-363, Jan. 1999
- [54] J. Nash, "Optimal Allocation of Tracking Resources," *Proc. IEEE Conf. Decision and Control*, New Orleans, LA, pp. 1177-1180, 1977.
- [55] D. Nicholson, C. M. Lloyd, S. J. Julier, and J. K. Uhlmann, "Scalable Distributed Data Fusion," *Proc. 5th Int. Conf. Information Fusion*, pp. 630-635, July 2002.
- [56] S. D. O'Neil and L. Y. Pao, "Multisensor Fusion Algorithms for Tracking," *Proc. American Control Conf.*, San Francisco, CA, pp. 859-863, June 1993.
- [57] Y. Oshman, "Optimal Sensor Selection Strategy for Discrete-Time State Estimators," *IEEE Trans. Aerospace and Electronic Systems*, Vol. 30, No. 2, pp. 307-314, April 1994.
- [58] L. Y. Pao, "Centralized Multisensor Fusion Algorithms for Tracking Applications," *Control Engineering Practice*, Vol. 2, No. 5, pp. 875-887, Oct. 1994.
- [59] L. Y. Pao, "Multisensor Multitarget Mixture Reduction Algorithms for Tracking," *J. Guidance, Control, and Dynamics*, Vol. 17, No. 6, pp. 1205-1211, Nov.-Dec. 1994.
- [60] L. Y. Pao and R. M. Powers, "A Comparison of Several Different Approaches for Target Tracking with Clutter," *Proc. American Control Conf.*, Denver, CO, pp. 3919-3924, June 2003.
- [61] C. Pohl and H. Touron, "Operational Applications in Multi-sensor Image Fusion," *Int. Archives of Photogrammetry and Remote Sensing*, Vol. 32, Part 7-4-3, Valladolid, Spain, June 1999.
- [62] C. Rago, P. Willett, and R. L. Streit, "A Comparison of the JPDA and PMHT Tracking Algorithms," *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, Vol. 5, pp. 3571-3574, May 1995.
- [63] B. Ristić, A. Farina, D. Benvenuti, and M. S. Arulampalam, "Performance Bounds and Comparison of Nonlinear Filters for Tracking a Ballistic Object on Re-entry," *IEE Proc. Radar, Sonar and Navigation*, Vol. 150, No. 2, pp. 65-70, April 2003.
- [64] S. I. Roumeliotis and G. A. Bekey, "Distributed Multirobot Localization," *IEEE Trans. Robotics and Automation*, Vol. 18, No. 5, pp. 781-795, Oct. 2002.
- [65] D. J. Salmond, "Mixture Reduction Algorithms for Target Tracking in Clutter," *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, Vol. 1305, pp. 434-445, 1990.
- [66] W. Schmaedeke, "Information-based Sensor Management," *Proc. SPIE*, Vol. 1955, 1993.
- [67] W. Schmaedeke and K. Kastella, "Event-averaged Maximum Likelihood Estimation and Information Based Sensor Management," *Proc. SPIE*, Vol. 2232, pp. 91-96, 1994.
- [68] W. Schmaedeke and K. Kastella, "Information Based Sensor Management and IMMKF," *Proc. SPIE Conf. Signal Processing of Small Targets*, Vol. 3373, Orlando, FL, pp. 390-401, 1998.
- [69] A. Sinha, Y. Bar-Shalom, D. W. Blair, and T. Kirubarajan, "Radar Measurement Extraction in the Presence of Sea-Surface Multipath," *IEEE Trans. Aerospace and Electronic Systems*, Vol. 39, No. 2, pp. 550-567, 2003.
- [70] M. I. Skolnik, *Introduction to Radar Systems*, Third Ed. McGraw-Hill, Boston, MA, 2001.
- [71] L. D. Stone, M. L. Williams, and T. M. Tran, "Track-to-Track Association and Bias Removal," *Proc. SPIE Conf. Signal Processing of Small Targets*, Vol. 4728, pp. 315-329, Orlando, FL, 2002.
- [72] R. Streit and T. Luginbuhl, "Maximum Likelihood Method for Probabilistic Multi-Hypothesis Tracking," *Proc SPIE Conf. Signal and Data Processing of Small Targets*, Vol. 2335, pp. 394-405, 1994.
- [73] L. Trailović and L. Y. Pao, "Computing budget allocation for optimization of sensor processing order in sequential multi-sensor fusion algorithms," *Proc. American Control Conf.*, Arlington, VA, pp. 1841-1847, June 2001.
- [74] L. Trailović and L. Y. Pao, "Variance Estimation and Ranking for Gaussian Mixture Distributions in Target Tracking Applications," *Proc. IEEE Conf. Decision and Control*, Las Vegas, NV, pp. 2195-2201, Dec. 2002.
- [75] L. Trailović and L. Y. Pao, "Position error modeling using gaussian mixture distributions with application to comparison of tracking algorithms," *Proc. American Control Conf.*, Denver, CO, pp. 1272-1277, June 2003.
- [76] L. Trailović and L. Y. Pao, "Computing Budget Allocation for Efficient Ranking and Selection of Variances with Application to Target Tracking Algorithms," *IEEE Trans. Automatic Control*, Vol. 49, No. 1, pp. 58-67, Jan. 2004.
- [77] M. Wada, K. S. Yoon, and H. Hashimoto, "Nonlinear Filter Road Vehicle Model Development," *Proc. IEEE Conf. Intelligent Transportation Systems*, pp. 734-739, Aug. 2001.
- [78] E. A. Wan and R. Van Der Merwe, "The Unscented Kalman filter for Nonlinear Estimation," *IEEE Symp. Adaptive Systems for Signal Processing, Communications, and Control, AS-SPCC 2000*, pp. 153-158, Oct. 2000.
- [79] G. A. Watson and W. D. Blair, "Revisit Control of a Phased Array Radar for Tracking Maneuvering Targets when Supported by a Precision ESM Sensor," *Proc. SPIE Conf. Signal Processing of Small Targets*, Orlando, FL, Vol. 2235, pp. 448-459, 1994.
- [80] G. A. Watson, W. D. Blair, and T. R. Rice, "Enhanced Electronically Scanned Array Resource Management through Multisensor Integration," *Proc. SPIE Conf. Signal Processing of Small Targets*, Vol. 3163, Orlando, FL, pp. 329-340, 1997.
- [81] D. Willner, C. B. Chang, and K. P. Dunn, "Kalman Filter Algorithms for a Multi-Sensor System," *Proc. IEEE Conf. Decision and Control*, pp. 570-574, Nov. 1976.
- [82] Z. Zhang and K. J. Hintz, "OGUPSA Sensor Scheduling Architecture and Algorithm," *Proc. SPIE Conf. Signal Processing, Sensor Fusion, and Target Recognition V*, Vol. 2755, pp. 296-303, 1996.