Interacting Multiple Model Methods in Target Tracking: A Survey

E. MAZOR
Technion, Israel Institute of Technology

A. AVERBUCH
Tel Aviv University

Y. BAR-SHALOM, Fellow, IEEE
University of Connecticut

J. DAYAN
Technion, Israel Institute of Technology

The Interacting Multiple Model (IMM) estimator is a suboptimal hybrid filter that has been shown to be one of the most cost-effective hybrid state estimation schemes. The main feature of this algorithm is its ability to estimate the state of a dynamic system with several behavior modes which can "switch" from one to another. In particular, the IMM estimator can be a *self-adjusting variable-bandwidth* filter, which makes it natural for tracking maneuvering targets. The importance of this approach is that it is the best compromise available currently between complexity and performance: its computational requirements are nearly linear in the size of the problem (number of models) while its performance is almost the same as that of an algorithm with quadratic complexity.

The objective of this work is to survey and put in perspective the existing IMM methods for target tracking problems. Special attention is given to the assumptions underlying each algorithm and its applicability to various situations.

Manuscript received August 15, 1995; revised May 24 and November 7, 1996; January 3, 1997.

IEEE Log No. T-AES/34/1/00182.

The work of Y. Bar-Shalom was supported by ONR Grant N00014-91-J-01950 and by AFOSR Grant F49620-95-1-0229.

Authors' addresses: E. Mazor and J. Dayan, Dept. of Mechanical Engineering, Technion, Israel Institute of Technology, Haifa 32000, Israel; A. Averbuch, School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel; Y. Bar-Shalom, Dept. of Electrical and Systems Engineering, U157, University of Connecticut, Storrs, CT 06269-2157, ybs@ee.ucomm.edu.

I. INTRODUCTION

Considerable research has been undertaken in the field of estimation theory in relation to the multitarget-multisensor tracking (MTMST) problem. This is of interest in both military and civilian applications. Ballistic Missile Defense and Airborne Surveillance require identification and tracking of several hundred targets in real time. A typical scenario includes maneuvering and nonmaneuvering targets as well as environmental clutter and false alarms produced by the sensor receiver noise and atmospheric disturbances. These problems occur in sophisticated weapon delivery systems, satellite surveillance systems, air defense, ocean/battlefield surveillance, air traffic control (ATC), and nonmilitary vehicle tracking systems. In addition, the techniques have general application to statistical pattern recognition and robotics, etc. The main purpose of a tracking system for traffic control or air defense is the estimation of target trajectories in the controlled area and their prediction into the near future. In a civilian ATC system, the estimated trajectories (tracks) are used to check the standard separation between pairs of targets for maintenance of safety conditions (collision avoidance) and regularity of traffic flow. In a maritime collision avoidance system, each ship monitors the traffic situation in order to detect the possibility of colliding targets. In an air defense system, the estimated trajectories are generally used to help perform some of the following functions [51]: threat identification, threat evaluation, weapon control and assignment, calculation of the predicted position (for fire or launch of missile), kill evaluation.

Although the mathematical structure of the optimal solution of the MTMST problem is well understood [7, 15, 18], the computational complexity of the optimal tracking algorithm (the Multiple Hypothesis Tracker (MHT)) limits its practical realization using even the largest and fastest computers available in the foreseeable future. Suboptimal implementations of the MHT have many potential pitfalls.[1] For this reason, many different tracking algorithms have been developed which sacrifice optimal performance for the sake of computational feasibility. Intuition suggests that some suboptimal approximations to the optimal algorithms result in little performance loss, but there exists no reliable quantitative analysis of suboptimal performance that demonstrates that this is, in fact, the case. The basic research problem is to find effective analytical techniques for the estimation problems. MTMST problems involve both the estimation of continuous-valued parameters such as target positions, velocities and accelerations,

[1] This was illustrated in [70] where an MHT (augmented with maneuver hypotheses in addition to data association hypotheses) kept losing a test flight target.

and the testing of discrete hypotheses, such as the association of measurement data with targets. Thus, it is natural to pose the problem of MTMST as a hybrid state estimation problem, that is, estimation for a partially observed stochastic process with discrete- and continuous-valued states. The hybrid state viewpoint [9, 38, 49, 85, 97] provides both a natural formulation for many types of surveillance and tracking problems and a powerful framework for deriving theoretically optimal and practical suboptimal tracking algorithms. Hybrid systems are characterized by the following.

1) State (consisting of kinematic and, possibly, feature components) that evolves according to a stochastic difference (or differential) equation model,

2) Model that is governed by a discrete stochastic process: it is one of a finite number of possible models (each corresponds to a behavior mode), that undergo jumps (switches) from one model (behavior mode) to another according to a set of transition probabilities (the dynamic Multiple Model (MM) approach).

The value of hybrid models for tracking algorithms is that the occurrence of target maneuvers can be explicitly included in the kinematic equations through regime jumps [33, 73]. The MM adaptive estimation approach is based on the fact that the behavior of a target cannot be characterized at all times by a single model, but a finite number of models can adequately describe its behavior in different regimes. Several approaches have been proposed to perform the state estimation of a system together with identification of each model (out of a finite set). Although static (i.e., nonswitching) MM estimation algorithms have been known since the mid 1960s, practical algorithms for the switching case have only recently become available. A unified treatment and a brief survey of the various suboptimal state estimation and system structure detection algorithms were presented in [93] for discrete systems with abruptly changing structure where the changes are modeled by a finite state Markov chain. The algorithms considered here assume that the transition probability matrix governing the Markov chain is known. The proposed approach assumes that the actual (true) system is one of finite number of possible models and that the a priori probability of each model is known. Modeling with Markov jump parameters may be thought of as an extension of the original static MM approach (or partitioning approach). In the former, Tugnait [93] allows switchings among the hypothesized models unlike the static MM approach where a candidate model is assumed to describe the true system behavior for all time. In several applications, switching parameter modeling is more realistic [85] because the underlying true system does have time-varying characteristics. Note that partitioning is a powerful approach resulting in parallel processing algorithms and robust solutions [72, 76, 92]. The case when the transition probability matrix is unknown deserves attention and has been considered by several authors for some restricted versions of the general problem [77, 93]. The case where jumps in the system structure are modeled by a semi-Markov chain was first discussed in [79].

A unified tutorial presentation on the topic of MM estimation and other related techniques from a target tracking oriented perspective can be found in [14, 15, 18]. Algorithms for hybrid state estimation in general consist of a bank of model-matched filter modules and some algorithm to organize the cooperation between the individual filters. The modules can be Kalman filters (KF) or other filters as discussed below. Compared with other MM approaches the main advantage of hybrid system-based solutions is that they are not subject to the so-called "obliviousness to detection" difficulty: the a priori jump probabilities used in the regime dynamics make hybrid algorithms alert to regime changes whereas classical solutions are biased toward the no-maneuver hypothesis after long quiescent periods [34]. One of the significant schemes is the so-called Generalized Pseudo-Bayesian (GPB) method [1, 40, 93] and the other is the Interacting Multiple Model (IMM) algorithm [28, 30]. The general structure of these algorithms consists of a bank of filters for the state cooperating with a filter for the parameters. A GPB algorithm of order k (GPBk) needs N_f^k filters in its bank [93], where N_f is the number of models in the bank and k is the number of sampling periods over which mode jumps are considered. The IMM algorithm is conceptually similar to GPB2 and performs nearly as well as the GPB2 method, but requires only N_f filters to operate in parallel, i.e., it has notably less computations, namely, the cost of GPB1 [30]. The IMM estimator, originally proposed by [28], is a suboptimal hybrid filter that was shown to achieve an excellent compromise between performance and complexity. Its complexity is nearly linear (with a small quadratic component) in the size of the problem (number of models) while its performance is almost the same as that of the GPB2. Another advantage (shared with the other MM estimators) is its modularity. It can be set up using KF or extended KF (EKF) as its building blocks to account for nonlinearities in the motion equation (e.g., for coordinated turns) and in the measurement equation for range-azimuthal-elevation observations, or probabilistic data association filters (PDAF) and joint PDAF (JPDAF), based on KF or EKF, if data association (due to false alarms or neighboring targets) is a significant problem.

One of the first applications of the IMM is the work of [33] for the European ATC (EUROCONTROL), based on the assumption that the governing system equations describe a Markov chain process. The state of this chain represents the mode of the target flight during a particular time interval. This mode is assumed to be among the possible modes of the set of all modal states at all times. The set of modes consists, for example, of a constant velocity model and several maneuver models. The process is characterized by its a priori transition probabilities [14], which constitute the transition matrix. These are design parameters for the algorithm. In the framework of the Hadamard project of the French Civil Administration, advanced tracking methods were investigated in [94]. The project dealt with the aircraft motion modeling and showed that stringent speed estimation can be obtained only with MM algorithms.

The IMM estimation algorithm has triggered a large variety of related approaches to target tracking. The goal of this survey paper is to provide an overview of these important developments. The organization is as follows. In Section II we describe the problem of hybrid state estimation which estimates the base state and the modal state based on the measurement sequence. In Section III we give detailed description of the IMM algorithm and its varieties. Comparison of the main MM algorithms is given in Section IV. Parallel implementation is discussed in Section V. Nonsimulation performance prediction techniques for IMM are the topic of Section VI. The application of an IMM estimator for a phased array radar is reviewed in Section VII. Finally, using IMM algorithms for target tracking with glint noise and noise identification is discussed in Section VIII.

II. PROBLEM FORMULATION

A general description for a hybrid system with additive noise is [65]

$$x(k+1) = f[k,x(k),m(k+1)] + g[k,x(k),m(k+1),v[k,m(k+1)]] \quad (1)$$

with mode-dependent noisy measurements

$$z(k) = h[(k,x(k),m(k)] + w[k,m(k)] \quad (2)$$

and a (possibly base state dependent) Markovian transition of the system mode

$$P\{m_j(k+1) \mid m_i(k)\} = \phi[k,x(k),m_i,m_j] \quad \forall \quad m_i,m_j \in M_s \quad (3)$$

where $x(\cdot)$ is the base state, $m(k)$ is the modal state (system mode index) at time k, which denotes the mode in effect during the sampling period ending at k, $P\{\cdot\}$ is probability, $m_j(k) \triangleq \{m(k) = j\}$ is the event that mode j is in effect at time k, M_s is the set of all modal states at all times, $v[\cdot]$ and $w[\cdot]$ are the mode-dependent process and measurement noise sequences with means $\overline{v}_j$ and $\overline{w}_j$ and covariances Q_j and R_j, respectively. It is assumed that v and w, uncorrelated with $x(0)$, are mutually uncorrelated[2] and that f, g, and h are known. Also $x(0)$ is assumed to be Gaussian with appropriate mean and covariance. All vectors and matrices are assumed to have appropriate dimensions. Note that (2) implies the availability of noisy, base state information only and that the mode information is embedded (hidden) in the measurement sequence. This description covers the case of nonstationary noises by modeling the noise sequences as jumping from one set of parameters to another one.

In a fixed-structure (or fixed mode-set) hybrid system, a set of modes must be selected in advance. Denoting by M_f this fixed set of N_f modes (this set might be smaller than M_s or, simply M_s is not really known), the system (1)–(3) is then approximated as

$$x(k+1) = f_j[k,x(k)] + g_j[k,x(k),v_j(k)] \quad \forall \quad j \in M_f \quad (4)$$

$$z(k) = h_j[k,x(k)] + w_j(k) \quad \forall \quad j \in M_f \quad (5)$$

with M_f "smaller" than M_s or just because M_s is not perfectly known, and jumps between the system modes are modeled by switching from one model to another, governed by the same law as given in (3). Here, N_f is the cardinality of the set M_f, i.e., the possible number of system modes, and subscript j denoted quantities pertaining to model (mode) m_j.

A jump linear fixed-structure hybrid system with mode transition modeled by a semi-Markov process, such as the state-dependent Markov chain, can be described by the equations [29, 31–33, 37–39, 78]

$$x(k+1) = F_j(k)x(k) + \Gamma_j(k)v_j(k) \quad \forall \quad j \in M_f \quad (6)$$

$$z(k) = H_j(k)x(k) + w_j(k) \quad \forall \quad j \in M_f \quad (7)$$

where Γ_j is the process noise gain matrix. The model switching process considered is of the semi-Markov type. The process is specified by a family of transition matrices $p_{ij}(\tau_i)$, where τ_i is the system's sojourn time in model i, i.e., it is a "sojourn-time-dependent Markov" (STDM) chain, which belongs to the semi-Markov class. The current probabilities of transition for the STDM process (chain) are defined as

$$p_{ij}(\tau) = P\{m_j(k+1) \mid m_i(k), \ \tau_i(k) = \tau\} \quad \forall \quad i,j \in M_f \quad (8)$$

where $\tau_i(k)$ is the sojourn time in state i at time k. It is assumed that at $k = 0$ the sojourn time (in whatever state the system model is) is $\tau = 1$. Thus, the values τ are taken from 1 to the maximum, which at time k is

[2] If the two noise sequences are correlated one can modify the estimator modules (see e.g., [14]).

then $k+1$. The process and measurement noises are Gaussian mutually uncorrelated with zero mean and known covariances.

The simplest hybrid system is one that can be described by a set of linear models as (6), (7) with the mode transition governed by a first-order homogeneous Markov chain

$$P_{ij}\{m_j(k+1) \mid m_i(k)\} = p_{ij} \qquad \forall \quad i,j \in M_f \tag{9}$$

where p_{ij} is the Markov transition probability from mode i to mode j. It is assumed, as before, that the process and measurement noises are Gaussian mutually uncorrelated with zero mean and known covariances.

When the Markovian switching coefficients can be isolated to a system bias, such linear hybrid system can be represented as [21]

$$x(k+1) = F(k)x(k) + \Gamma_j(k)b(k) + \Gamma^x(k)v^x(k) \tag{10}$$

$$b(k+1) = D_j(k)b(k) + \Gamma_j^b(k)v^b(k) \tag{11}$$

$$z(k) = H(k)x(k) + G_j^b(k)b(k) + w(k) \tag{12}$$

with bias transition governed by a finite state Markov chain

$$P_{ij}\{m_j^b(k+1) \mid m_i^b(k)\} = p_{ij} \qquad \forall \quad i,j \in M_f \tag{13}$$

where $b(k)$ is the system bias, p_{ij} is the Markov transition probability from bias model m_i^b to bias model m_j^b; $v^x(k)$ and $v^b(k)$ are white Gaussian errors for the system state and bias process respectively; $w(k)$ is the white Gaussian error in the measurement and independent of $v^x(k)$ and $v^b(x)$. It is assumed that covariances of $v^x(k)$, $v^b(k)$ and $w(k)$ are known ($Q^x(k)$, $Q^b(k)$ and $R(k)$ respectively); the functions $F(k), \Gamma_j(k), \Gamma^x(k)$ are known also.

The problem of hybrid state estimation is to estimate the base state and the modal state based on the (noisy base state) measurement sequence.

Following [33] the outline principles of the track maintenance system are as follows. Once a track has been initiated, the track maintenance system updates that track as long as it is observed by at least one sensor. Hence, a multisensor track-continuation problem is reduced to a "single-sensor" problem, where the updating is sequential across the sensors. Track continuation has to cope with several uncertainties, of which the following four cause the major difficulties:

1) nonlinear target dynamics during a turn,
2) sudden starts and stops of maneuvers (mode switching),
3) the association of measurements with existing tracks,
4) Gaussian-mixture-type measurement noise.

In developing a high-quality track continuator the Bayesian approach is employed. It consists of building appropriate stochastic models, developing the exact filter solution from the Bayesian estimation theory for Markov processes, and finally introducing efficient numerical approximations of the exact solution. By following these three steps, several approximate Bayesian methods were developed for each of the above subproblems. The various approximate Bayesian methods show a trade-off between computational load and performance, depending on the particular application.

To cope with the first difficulty, the well-known extended Kalman method of linearization along the predicted path is adopted as a good compromise.

To cope with the second difficulty, the aircraft behavior is modeled as a hybrid state Markov process. An aircraft trajectory can be subdivided into distinct segments, corresponding to modes of flight. An appropriate model for the switching between modes is a finite-state (semi-) Markov process. An aircraft is then modeled as having a hybrid state; consisting of a continuous Euclidean-valued (diffusive) state component and a discrete-valued state component. The discrete state describes the mode of flight. The diffusive state describes the horizontal position, ground speed, course, and, for some modes, bank angle, flight-path angle, and transversal acceleration. The evolution in time of the discrete state (i.e., switching between modes) and the diffusive state are determined by the physical relations between the state components, the jump-type changes of the control commands of the pilot, and disturbances caused by both the pilot and the wind. In this model, switching from one mode to another generally causes a simultaneous random jump in bank angle, flight-path angle, and acceleration. The combination of discrete and diffusive states in a maneuvering trajectory model causes the associated tracking problem to fall within the class of so-called hybrid-state estimation problems.

From the various approximate methods to cope with the third difficulty, an MM probabilistic data association (PDA) method [7, 15, 54] is judged to be an excellent compromise to update an aircraft track from new observations. Moreover, the PDA approach allows to include an efficient solution to the fourth difficulty (outliers).

Efficient cooperation between the KFs is realized by an interaction between the estimates (the conditional mean and covariance given a particular mode is in effect) for the different modes at the beginning of each filter cycle. The interaction is determined by the conditional probabilities of switching between modes. For problems like tracking, the IMM interaction is so effective, that IMM algorithm performs almost like the exact Bayesian filter [30]. Moreover, the IMM requires a far lower computational power than other high-performance algorithms for tracking maneuvering aircraft [8, 30, 34].

III. THE IMM ESTIMATOR AND ITS VARIETIES

The IMM estimator has been shown to be one of the most cost-effective and simple schemes for the estimation in hybrid systems [5, 10–12, 15, 55, 58] and therefore it is suitable to be used in MTMST. The IMM design parameters are as follows.

1) The set of models for the various regimes and their structure.

2) The process noise intensities for the various models, in particular the nonmaneuvering model with low-level process noise and the maneuvering model(s) with certain higher noise levels determined by the assumed maneuverability of the targets.

3) The jump structure (usually Markov) and the transition probabilities between the models from the selected set. These probabilities are chosen according to the designer's beliefs about the frequency of the regime switches and can be subsequently adjusted based on Monte Carlo simulation results. When the transition probabilities of the Markov chain are zero the IMM algorithm reduces to the well-known static MM algorithm.

The IMM algorithm has three desirable properties: it is recursive, modular, and has fixed computational requirements per cycle. In each cycle it consists of three major steps: interaction (mixing), filtering, and combination. At each time, the initial condition for the filter matched to a certain mode (a module) is obtained by mixing the state estimates of all filters at the previous time under the assumption that this particular mode is in effect at the current time. This is followed by a regular filtering (prediction and update) step, performed in parallel for each mode. Then, a combination (weighted sum) of the updated state estimates of all filters yields the state estimate. The probability of a mode being in effect plays a key role in the weighting of the mixing and the combination of states and covariances.

The main feature of this algorithm is its ability to estimate the state of a dynamic system with several behavior modes which can "switch" from one to another. In particular, the IMM estimator can be a *self-adjusting variable-bandwidth* filter, which makes it natural for tracking maneuvering targets.

The combinations of the IMM approach with PDAF and JPDAF methods for single and multiple targets in clutter are discussed extensively in [9, 11, 13, 33, 48, 49, 58, 63]. These recursive algorithms have fixed computational and memory requirements and can initiate tracks, maintain tracks in the presence of maneuvers, and terminate tracks if warranted. The algorithms are useful for situations of low signal-to-noise ratios (SNRs), where the detection threshold has to be set low to detect the targets, this leading to a high rate of false alarms for which logic-based techniques are not adequate. These algorithms, which yield model probabilities, provide the true target probability for each track under consideration. Thus, these algorithms can assess their own reliability (track loss); that is, they can be called "intelligent." A number of track formation examples with the IMMPDAF on real data can be found in [15].

A. Baseline IMM Algorithm

The baseline IMM algorithm [8, 14, 28, 30] assumes the simplest form of hybrid system (6), (7), (9); each mode-matched filter is a standard KF.

Notations: Z^k denotes the measurement sequence through time k; $\hat{x}(i \mid l)$ denotes the state estimate at time i conditioned on Z^l, and $P(i \mid l)$ is the associated covariance matrix; quantities pertinent to mode j are denoted with subscript j; $N(y;\bar{y},P)$ denotes the (multivariate) Gaussian density function of y with mean $\bar{y}$ and covariance P; $\sum_i$ stands for $\sum_{m_i \in M_f}$; $\hat{x}_j(k \mid k)$ and $P_j(k \mid k)$ are the state estimate and its covariance in mode-matched filter j at time k; $\hat{x}_{0j}(k \mid k)$, $P_{0j}(k \mid k)$ are the mixed initial condition for mode-matched filter j at time k; $\hat{x}(k \mid k)$, $P(k \mid k)$ are the combined state estimate and its covariance; $\mu_j(k)$ is the mode probability at time k; $\mu_{i|j}(k \mid k)$ is the mixing probability at time k (the weights with which the estimates from the previous cycle are given to each filter at the beginning of the current cycle); $\Lambda_j(k)$ is the likelihood function of mode-matched filter j.

The basic IMM algorithm (one cycle) is as follows.

Interaction. $\forall\, i,j \in M_f$, $\mu_{i|j}(k-1 \mid k-1) = (1/\bar{c}_j)p_{ij}\mu_i(k-1)$ (mixing probability), where $\bar{c}_j$ is a normalization factor

$$\bar{c}_j = \sum_i p_{ij}\mu_i(k-1)$$

$$\hat{x}_{0j}(k-1 \mid k-1) = \sum_i \hat{x}_i(k-1 \mid k-1)\mu_{i|j}(k-1 \mid k-1)$$

$$\begin{aligned} P_{0j}(k-1 \mid k-1) = \sum_i \{ & P_i(k-1 \mid k-1) \\ & + [\hat{x}_i(k-1 \mid k-1) - \hat{x}_{0j}(k-1 \mid k-1)] \\ & \times [\hat{x}_i(k-1 \mid k-1) - \hat{x}_{0j}(k-1 \mid k-1)]^T \} \\ & \times \mu_{i|j}(k-1 \mid k-1). \end{aligned}$$

Filtering. $\forall\, j \in M_f$

$$\begin{aligned} \hat{x}_j(k \mid k-1) &= F_j(k-1)\hat{x}_{0j}(k-1 \mid k-1) \\ &\quad + \Gamma_j(k-1)\bar{v}_j(k-1) \\ P_j(k \mid k-1) &= F_j(k-1)P_{0j}(k-1 \mid k-1)F_j(k-1)^T \\ &\quad + \Gamma_j(k-1)Q_j(k-1)\Gamma_j(k-1)^T \end{aligned}$$

$$\hat{x}_j(k \mid k) = \hat{x}_j(k \mid k-1) + W_j(k) r_j(k)$$

$$P_j(k \mid k) = P_j(k \mid k-1) - W_j(k) S_j(k) W_j(k)^T$$

$$r_j(k) = z(k) - \hat{z}_j(k \mid k-1) \quad \text{(residual)}$$

$$\hat{z}_j(k \mid k-1) = H_j(k)\hat{x}_j(k \mid k-1)$$

$$\text{(measurement prediction)}$$

$$S_j(k) = H_j(k) P_j(k \mid k-1) H_j(k)^T + R_j(k)$$

$$\text{(residual covariance)}$$

$$W_j(k) = P_j(k \mid k-1) H_j(k)^T S_j(k)^{-1} \quad \text{(filter gain)}$$

$$\Lambda_j(k) = N(r_j(k); 0, S_j(k)) \quad \text{(likelihood function)}$$

$$\mu_j(k) = \frac{1}{c}\Lambda_j(k)\sum_i p_{ij}\mu_i(k-1)$$

$$= \frac{1}{c}\Lambda_j(k)\bar{c}_j$$

(mode probability, c is a normalizing factor).

Combination. $\forall\, j \in M_f$

$$\hat{x}(k \mid k) = \sum_j \hat{x}_j(k \mid k)\mu_j(k)$$

$$P(k \mid k) = \sum_j \{P_j(k \mid k) + [\hat{x}_j(k \mid k) - \hat{x}(k \mid k)] \times [\hat{x}_j(k \mid k) - \hat{x}(k \mid k)]^T\}\mu_j(k).$$

A second-order dependence of the base state on the modal state was considered in [22, 29, 33] and designated as "generalized IMM". With the notations of (1), this is written as

$$x(k+1) = f[k, x(k), m(k+1), m(k)] + g[k, x(k), m(k+1), m(k), v[k, m(k+1), m(k)]].$$

With the complexity of the (slightly modified) IMM estimator being the same, this formulation allows a different state equation at the instant of the jump, i.e., when $m(k+1) \neq m(k)$, without the need for an extra model. Furthermore, it can operate without predefined acceleration levels. It was indicated in [94] that an IMM with second-order dependency yielded lower errors than a conventional IMM. Smoothing versions of the IMM have been studied in [31, 56].

B. STDM-Based IMM Estimator

In this approach [29, 31–33, 37–39, 78] the hybrid system is modeled by the equations (6)–(8); each mode-matched filter is a standard KF. Each model transition probability is a known function of the sojourn time given by (8) and has a sojourn time $\tau_i(k)$ in state i which is, however, not known. The conditional probability mass function (pmf) of the sojourn time in state $m(k) = i$ based on the available information Z^k at time k, is

$$\begin{aligned} g_i^k(\tau) &\triangleq P\{\tau_i(k) = \tau \mid m(k) = i, Z^k\} \\ &= P\{\tau_i(k) = \tau \mid m(k) = i, Z^{k-1}\} \\ &= P\{m(k-1) = i, \ldots, m(k-\tau+1) \\ &\qquad = i, m(k-\tau) \neq i \mid m(k) = i, Z^{k-1}\} \end{aligned} \tag{14}$$

where the perfect knowledge of the state $m(k)$ allows one to go down to one index less in the conditioning, i.e., Z^{k-1}. The conditional pmf of the sojourn time τ in state i at time k (14) is given by the expressions in [37, 39]. Following (8) the conditional probability of transition from i to j at time $k-1$, given the observations Z^{k-1} is, in terms of (14),

$$\begin{aligned} \hat{p}_{ij}(k-1) &\triangleq P\{m(k) = j \mid m(k-1) = i, Z^k\} \\ &= \sum_{\tau=1}^{k} p_{ij}(\tau) g_i^{k-1}(\tau). \end{aligned} \tag{15}$$

Note that the argument of p_{ij}, defined in (8), is the sojourn time, while the argument of $\hat{p}_{ij}$, defined above is the current time. The filter has access only to the observations from which the conditional pmf of the sojourn time can be obtained: this, in turn, is to be used in calculation of the conditional transition probabilities (15).

There are numerous examples of systems which have discrete models that randomly vary with time and experience switchings, between different models, after a random sojourn time. In some situations the switching probabilities depend on the sojourn time. Such situations are commonly experienced in MTMST [7, 15, 18, 35, 96]. The model switching process is characterized by a family of transition probability matrices $p_{ij}(\tau_i)$, i.e., it is STDM chain, which belongs to the semi-Markov class. The example in [37, 39] shows that STDM-based IMM estimator gives a stable estimation filter for both two- and three-model cases. The three-model filter was shown to give a small probability of error in system structure detection and thus the true system model becomes rapidly known, which is critical in failure detection schemes.

Summary of the STDM-based IMM estimator (one cycle) is as follows.

Interaction. $\forall\, i, j \in M_f$

$$\mu_{i|j}(k-1 \mid k-1) = \frac{1}{\bar{c}}\hat{p}_{ij}(k-1)\mu_i(k-1)$$

$$\text{(mixing probability)}$$

The conditional probability of transition from i to j at time $k-1$, given the observations Z^{k-1} is

$$\hat{p}_{ij}(k-1) = \sum_{\tau=1}^{k} p_{ij}(\tau) g_i^{k-1}(\tau)$$

$$\bar{c}_j = \sum_i \hat{p}_{ij}(k-1)\mu_i(k-1).$$

$\hat{x}_{0j}(k-1 \mid k-1)$ and $P_{0j}(k-1 \mid k-1)$ are defined in Section IIIA.

Filtering. From Section IIIA

$$\mu_j(k) = \frac{1}{c}\Lambda_j(k)\sum_i \hat{p}_{ij}(k-1)\mu_i(k-1) = \frac{1}{c}\Lambda_j(k)\bar{c}_j.$$

Combination. See Section IIIA.

More work on semi-Markov switching systems can be found in [32].

C. IMM Algorithms for Systems with Markovian Switching Coefficients Isolated to a System Bias

The idea of using a two-stage filter to implement an augmented state filter was introduced in [53], where the bias vector is a part of the augmented system state. The central filter is decoupled into two parallel filters. The first filter, the "bias-free" filter, is based on the assumption that the bias is nonexistent. The second filter, the bias filter, produces an estimate of the bias vector. The output of the first filter is then corrected with the output of the second. If the bias is deterministic and constant but unknown, the two-stage filter is equivalent to the augmented state filter [53]. As proposed in [3], the two-stage Kalman estimator is equivalent to the augmented state filter under an algebraic constraint on the correlation between the state process noise and the bias process noise. An important problem is the estimation of a linear system with Markovian switching coefficients. When the Markovian switching governs the system bias and the IMM algorithm is used for state estimation, the "bias-free" portion of the system state must be duplicated in each model and its corresponding filter [2, 3]. Such a linear system can be represented as in equations (10)–(13).

A novel approach to state estimation for systems with Markovian switching system bias was presented in [21]. This Interacting Multiple Bias Model (IMBM) algorithm integrates the baseline IMM algorithm and the recent developments of Alouani and Blair in two-stage estimation. The IMBM algorithm consists of a filter for the bias-free portion of the state model, a filter for each bias model, a model probability evaluator for each bias model filter, an estimate mixer at the input of the bias filters, and an estimate combiner at the output of the bias 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 bias estimate for each bias filter. The bias-free filter error covariance is also modified to reflect the spread of the means of the bias filters. At the beginning of filtering cycle, each bias filter uses a mixed bias estimate and the bias-free measurement residual to compute a new estimate of the bias and a likelihood for the bias model. The likelihoods, prior model probabilities, and the model switching probabilities are then used to compute new model probabilities. The overall bias estimate and compensated state estimate are then computed with the output of the bias-free filter, the outputs of the bias filters, and their model probabilities.

The IMBM algorithm was applied to tracking maneuvering targets, where the target acceleration was treated as a system bias with Markovian switching coefficients. This algorithm was called the Interacting Multiple Acceleration Model (IMAM) algorithm. In the IMAM algorithm the bias-free filter corresponds to the constant velocity filter, while the bias filters correspond to a zero acceleration model and two constant acceleration models. The constant acceleration models will have different process noises covariances, while one constant acceleration filter will utilize the kinematic constraint for constant speed targets [2, 3]. Thus, this IMAM algorithm includes a constant velocity model and three bias models. The first bias model, $m_1(k)$, which corresponds to a zero mean and zero covariance bias, does not require a separate filter. The second bias model, $m_2(k)$, corresponds to a constant state with the kinematic constraint, while the third bias model, $m_3(k)$, corresponds to a constant state. This IMAM algorithm which utilizes a zero bias model, $m_1(k)$, and two constant acceleration models is presented in [21]. The tracking performance of the IMAM algorithm is compared with the tracking performance of a similar baseline IMM algorithm. Simulation results show that IMAM provides tracking that is very similar to that provided by the baseline IMM algorithm, while requiring about 43% of the computations of the IMM algorithm, when a constant velocity model and two constant acceleration models are used. The computational requirements of the IMAM algorithm may allow MM algorithms to be implemented for target tracking in existing systems, where the IMM algorithm cannot be justified from a computational standpoint.

A special case of the IMAM algorithm is presented in [98]. This algorithm was called the Interacting Acceleration Compensation (IAC) algorithm. It was suggested that this IAC algorithm can further reduce the computational cost of the IMM algorithm while maintaining similar performance. The IAC algorithm incorporates the concept of the IMM algorithm for two motion models into the framework of the two-stage estimator. In the IAC algorithm, the two-stage estimator is viewed as having two acceleration models. The first acceleration model, $m_1(k)$, corresponds to the zero acceleration of the constant velocity model and does not require a filter (as in IMAM) and the second acceleration model, $m_2(k)$, is a constant state model. The IMM algorithm is used with the acceleration models to compute the acceleration estimate that compensates the constant velocity filter estimate. The tracking performance of the IAC algorithm is compared with an IMM algorithm containing a constant velocity model and a constant acceleration model. The IAC and IMM algorithms were simulated with a radar tracking system measuring the target range, bearing, and elevation at periodic intervals of 1.0 s. Simulation results showed that the tracking performance of the IAC algorithm approaches that of a comparable IMM algorithm, while requiring approximately 50% of the computations. The tracking performance of the IMM algorithm has been found to be better than the IAC algorithm when the targets maneuver through constant-speed turns and the data rates are low (i.e., four or five measurements per maneuver). As data rates are increased, the tracking performance of both algorithms becomes similar. The performance of the IAC algorithm may be improved by using a turning rate model to estimate the target acceleration for constant speed maneuvers by treating the acceleration as a dynamical bias [3]. The turning rate model would improve filter response during a maneuver by providing improved prediction over the constant velocity model. The use of turning rate models or the use of a kinematic constraint in the IMM algorithm for such targets has proved very effective. Multiple acceleration models have been incorporated into the bias or acceleration estimation in the IMBM algorithm [21]. The use of multiple acceleration models (IMAM algorithm) will aid in responding to target maneuvers. Since the IAC algorithm is most applicable for suddenly accelerating targets, the use of multiple acceleration models would improve the estimation of targets performing slight maneuvers.

D. IMM Estimators with Mode-Set Adaptation

Often, for practical problems an algorithm using a fixed set of a small number of models cannot yield accurate results [65, 69]. Apart from the increase in computation, use of more models does not guarantee better performance—actually, it may yield even poorer results. The major reason for the poor performance of the fixed "full" mode-set algorithms in some cases is that when a large number of modes are used at any time, many of them are so different from the system mode in effect at a particular time, that not only the computation for the filters matched to these modes provides no gain but also the unnecessary "competition" among modes degrades the performance. This degradation arises from the fact that the same (large) mode set is used for all time periods. To find a way out of this dilemma, [65] introduced the concept of variable structure, i.e., adaptation of the model set. The work [65] proposed several variable-structure algorithms, in contrast to the existing efforts of developing better implementable versions of the so-called optimal (fixed-structure) estimator. One solution is to use a time-varying mode set. That is, M_f in (4)–(5) is replaced by a time-varying set $M(k)$, to be determined from all the information available—in particular, the sequence of the measurements. This is further detailed in [69].

The "full mode-set" estimators described above have been considered to yield the performance bound of all IMM estimators. Unfortunately, these "optimal" estimators may give unsatisfactory results in some cases because they are the best only within the class of *fixed*-structure algorithms. By using a variable structure, it is possible to develop algorithms that can outperform these estimators.

The variable-structure (or adaptive mode-set) estimator can be viewed as a two-level hierarchical algorithm: MM at the lower level and multiple mode-set at the higher level. As was shown in [65, 69], the fixed-structure estimator is only a special case of the variable-structure estimator using a trivial partition consisting of the following single sequence of mode sets

$$M^k = \{M_f, M_f, \ldots, M_f\} \tag{16}$$

where M^k denotes a sequence of the mode sets through time k. That is, the variable-structure estimator reduces to one of the fixed-structure estimators if only sequences of fixed mode sets are allowed. Note that (16) is not the only possible partition. Suppose $M_s = \{1,2,3,4\}$. The following two sequences of mode sets constitute one of the many possible partitions at time $k = 2$:

$$M_1^2 = \{\{1,2,3,4\},\{1,2\}\}$$

$$M_2^2 = \{\{1,2,3,4\},\{3,4\}\}.$$

Thus, the choice of the partition can be made after the arrival of the measurements. The full-fledged variable-structure estimator is too complicated to be feasible. Nevertheless, it suggests the use of time-varying mode-set sequences since such sequences can lead to better results than a fixed sequence. The number of mode histories in the adaptive mode-set algorithm at k is $\prod_{t=1}^{k} N(t)$ ($N(t)$ is the number of modes at time t), which is much smaller than N_f^k since $N(t) < N_f$. Thus, the adaptive mode-set approach can provide a tremendous saving in computation over a fixed mode-set approach, apart from superior performance. Meanwhile, the same management techniques of mode histories as in the fixed mode-set algorithms can be retained. Practical suboptimal algorithms with variable structure are presented in [69]. A graph-theoretic formulation of MM estimation which leads to a systematic treatment of model-set adaptation and opens up new avenues for the study and design of the IMM estimation algorithms was also given.

An ad-hoc variable-structure method named Selected Filter IMM (SFIMM) has been proposed in [71]. The SFIMM algorithm is a modified IMM algorithm which uses a subset of filters with the specific subset chosen using decision rules. Particular attention was focused on the design parameters, namely, the subset type and the decision rule in this study. The coverage of thirteen filters was investigated. Two different subset types were examined. In subset type 1, suggested in [65], each subset "connects" to another one. This was found to have a higher possibility of shift and so when shifts occur a larger portion of models needed initialization and larger tracking errors ensued. Subset 2, also suggested in [65], used a symmetric structure, with some small overlapping, and yielded better tracking performance. Two different decision rules were used to decide whether a shift was required or not, first the maximum likelihood (ML) criterion which checked the most likely among all the models in the same subset to make a decision. In the IMM algorithm the likelihood function for each model has some contributions from all the models in the same subset, so it is a mixed likelihood. The second criterion was the maximum a posteriori (MAP) criterion which included the prior probability. Again, as the probability of each model has some contributions from all the models in the same subset, the a posteriori probability also has mixed terms. Since more information is used in this decision rule, the tracking performance was improved. The simulation results showed that the performances for all four designs of the SFIMM algorithm considered were acceptable, although design 4 which used subset type 2 and the MAP decision rule had the best performance. The tracking accuracy was only slightly worse than that of an IMM algorithm with thirteen subfilters and took approximately one quarter of the computation time.

Another approach, suggest by [82], is to use the so-called adaptive interacting multiple model (AIMM) algorithm. Each model is assigned a fixed deterministic acceleration to cope with the different target accelerations. A rough estimate of the target acceleration is taken from a biased filter which uses combined information from the models. This estimated acceleration is then taken into account in the models. Different maneuvering target scenarios were generated to compare the performance of the AIMM algorithm with the IMM algorithm. The performance of the AIMM algorithm in tracking is better than the IMM algorithm, which uses submodels with different noise levels, when accuracy and computation time are used in the assessment. A major advantage of the AIMM algorithm is that it does not need predefined fixed deterministic accelerations for the submodels. The IMM algorithm may provide better accuracy when one of the submodels is well matched with the motion of the target. However, the AIMM algorithm can provide better coverage of variable target accelerations using fewer submodels.

E. IMM Algorithm with Correlated Measurement Noise

In tracking airborne or missile targets using radar data, the measurement noise is significantly correlated when the measurement frequency is high. In the paper [55], a simple decorrelation process is proposed to enhance the IMM algorithm to track a maneuvering target with correlated measurement noise. It assumed the simplest form of hybrid system (6), (7), (9) and that the noise can be modeled as a first-order Markov process given by

$$w_j(k+1) = \lambda w_j(k) + w_j^0(k) \tag{17}$$

where λ is the correlation coefficient ($0 \leq \lambda \leq 1$) between successive measurements and w_j^0 is a white sequence.[3] To decorrelate the measurement noise a new measurement $z_a(k)$, denoted as "artificial measurement" in [90], is generated. Let $\overline{\lambda}$ be the estimated value of noise correlation, then the measurement equation can be rederived as follows

$$z_a(k) = z(k) - \overline{\lambda} z(k-1) \tag{18}$$

$$\begin{aligned}
&= (H_j x(k) + w_j(k)) - \overline{\lambda}(H_j x(k-1) + w_j(k-1)) \\
&= [H_j x(k) - \overline{\lambda} H_j (F_j)^{-1}(x(k) - \Gamma_j v_j(k-1))] \\
&\quad + [w_j(k) - \overline{\lambda} w_j(k-1)] \\
&= [H_j - \overline{\lambda} H_j (F_j)^{-1}] x(k) + \overline{\lambda} H_j (F_j)^{-1} \Gamma_j v_j(k-1) \\
&\quad + [(\lambda - \overline{\lambda}) w_j(k-1) + w_j^0(k)].
\end{aligned} \tag{19}$$

Let

$$\hat{H}_j = H_j - \overline{\lambda} H_j (F_j)^{-1} \tag{20}$$

$$\hat{\Gamma}_j = \overline{\lambda} H_j (F_j)^{-1} \Gamma_j \tag{21}$$

$$\hat{w}_j(k) = \hat{\Gamma}_j v_j(k-1) + (\lambda - \overline{\lambda}) w_j(k-1) + w_j^0(k) \tag{22}$$

Then

$$z_a(k) = \hat{H}_j x(k) + \hat{w}_j(k). \tag{23}$$

If $\overline{\lambda} \approx \lambda$ the new measurement noise $\hat{w}_j(k)$ would be white, but it is correlated with the process noise

[3] The notations in [55] were erroneous.

$v_j(k-1)$. By reformulating the dynamic equation (6) properly, the process noise can be made to be uncorrelated with new measurement noise [14]. In some practical systems, this procedure can be omitted with little degradation in performance since the term $\hat{\Gamma}_j v_j(k-1)$ is usually small. Thus, the baseline IMM algorithm can be applied to the case with correlated measurement noise by the following substitutions:

$$H_j \to \hat{H}_j \qquad w_j(k) \to \hat{w}_j(k)$$
$$z(k) \to z_a(k) \qquad \text{for} \quad j = 1,2,\ldots,N_f.$$

The results of computer simulations [55] indicated that the decorrelation process may improve system performance significantly, especially in the velocity and acceleration estimates. These large improvements in velocity and acceleration are particularly useful in situation when the target is suddenly accelerated by the pilot or missile guidance program.

F. Multisensor IMM Tracking Algorithms

In some military and civilian applications, the data are collected by a network of sensors distributed over a large geographic region. In such distributed sensor networks (DSNs), because of considerations such as reliability, survivability, and communication bandwidth, centralized processing is either undesirable or infeasible. Instead, the sensors supply data to a set of local processors/nodes which are connected by a communication network. The nodes process the local sensor data and exchange processing results with other nodes. The key problem in multitarget tracking is data association, i.e., the association of the measurements from single or multiple sensors to the set of targets. Many different approaches have been proposed for data association in a centralized framework [41], in which all the sensor measurements are transmitted to a centralized site where they are processed. One such approach is JPDAF (for multiple targets in clutter) and PDAF (for a single target in clutter); see [15]. In these algorithms the data association problem is handled by computing the posterior probabilities of the measurement associations with targets in clutter. These probabilities are used to update a set of Gaussian distributions representing (suboptimal) Bayesian estimates of the target locations. Several distributed tracking algorithms for DSN have been discussed in [41]. The paper [44] presents a distributed tracking scheme based on the JPDAF where both data association and distributed estimation are considered. In this scheme, each node first performs the tracking functions with the JPDAF using the local measurements and sends the processed results to other nodes. The receiving node then fuses the information from other nodes with its local information to arrive at a better estimate (sequential fusion). The paper [41] presents a JPDAF in DSN structure, where each local node processes the local sensor measurements and sends the local estimates periodically to the global processor (fusing node, parallel fusion). After processing the results, the global processor sends back the global estimates to each local node.

In many multisensor systems the number and type of sensors supporting a particular target track can vary with time due to the mobility, type, and resource limitations of the individual sensors. This variability in the configurations of the sensor system poses a significant problem when tracking maneuvering targets because of the uncertainty in the target motion model. In the paper [27], the IMM estimator is applied to the problem of tracking maneuvering targets with multiple, possibly intermittent, sensors. For the IMM estimator, the following two models were chosen: 1) the nearly constant velocity model with piecewise constant acceleration errors, and 2) constant acceleration model with piecewise constant acceleration increment errors. The IMM algorithm provides significantly better tracking performance than a nearly constant velocity KF (this filter is based on the assumption that the target is moving with constant velocity plus zero-mean, white Gaussian acceleration errors that are piecewise constant) even when the latter runs at a higher rate.

In the case where "track-to-track" fusion [15] is carried out, it is very important that the state estimators have realistic (consistent) covariances [14]. While the KF cannot be consistent since it uses only one model, the IMM has been shown to be nearly consistent. Lack of consistency of the KF can be a major problem in multisensor fusion.

Sequential Multisensor IMMPDAF Algorithm: The sequential multisensor (MS) IMMPDAF algorithm is presented in [15, 58]. This algorithm assumes the simplest form of hybrid system (6), (7), (9); each mode-matched filter is a standard PDAF [15]. The sensors (two—one infrared and one radar) are in the same location, assumed, for simplicity, synchronized and with the same sampling rate. Track initialization is assumed to have been made. At each scan, a validation gate, centered around the predicted measurement of the target, is set up for each sensor for the next scan.

The summary of the sequential MS IMMPDAF algorithm (one cycle) is as follows.

1) Mixing of the estimates from the previous time (see Section IIIA).

2) Predicted states and predicted measurements.

3) Measurement validation from sensor 1. The validation region has to be taken the same for each of the models as the largest of them.

4) Estimation with sensor 1 measurements in each filter. At the end of the estimation, one computes

a new "predicted" measurement for sensor 2, as in step 2.

5) Measurement validation from sensor 2. This step is similar to step 3.

6) Estimation with sensor 2 measurements in each filter. This step is similar to step 4.

7) Updating of the model probabilities. (See Filtering in Section IIIA).

8) Combination of the model-conditioned estimates. (See Combination in Section IIIA).

Parallel Multisensor IMMPDAF Algorithm: The parallel MS IMMPDAF algorithm is presented in [42, 43]. This algorithm assumes the simplest form of hybrid system (6), (7), (9) too; each mode-matched filter is a PDAF. The two-node scenario similar to that given in [41] is considered, where each node processes the local measurements from its own sensor and sends the local estimates to the fusion processor periodically. The fusion processor then sends back the processed results after each communication time. Lossless communication is assumed and the information communicated is the sufficient statistics. For each local node, the centralized IMMPDAF algorithm, where all measurements are sent to and processed with one processor, is described similar to baseline IMM algorithm with a standard PDAF. The goal is to compute the conditional state distribution given the local accumulated measurements. With the local conditional pdf of node i, one can derive the fusion algorithm for the baseline IMM filter. The information needed to be communicated from local nodes to the fusion node consists of a) the model probabilities; b) the association event probabilities; and c) the corresponding probability density functions (pdfs) (mean and covariance for Gaussian case). Three different scenarios were tested. First, each sensor tracks the target independently using the IMM algorithm from Section IIIA. Second, the measurements from sensors were first concatenated [100] into one, then the standard IMM algorithm, as in the first case, was applied. Finally the *distributed* (parallel MS IMMPDAF) case was simulated. Simulation results show that the performance of the distributed case depends explicitly on the amount of information being communicated between nodes. The results of centralized processing scheme represent the upper bound of the performance. When two nodes communicate every scan, the results of the distributed case are the same as the centralized case (but has the advantages of increased reliability), which confirms their theoretical equivalence. When two nodes communicate every two scans, the results of the distributed case are better than that of the decentralized case (no communication), which shows the advantage of interchanging information.

For a centralized MS PDAF the sequential update across sensors is preferable over the simultaneous update with the measurements from the various sensors (assumed synchronized) [84]. The reason the sequential update is superior is that it reduces the uncertainty faster and allows for better data association.

G. Square Root Algorithms for IMM Filters

Theoretically, the KF gives the unbiased, minimum variance estimate of the state vector of a linear dynamic system, disturbed by additive white noise when measurements of the state vector are linear, but corrupted by Gaussian white noise. Such performance is hardly ever realized in actual practice since the information required to construct the KF is only approximately known. The noise parameters and models may be based upon only relatively few data points, computer round-off errors may be significant, and the system model may not be adequate. The conventional KF algorithm is not numerically robust due to round-off errors and ill-conditioning problems. Numerical studies in the past [16, 62] have shown that the Kalman algorithm is so sensitive to computer round-off errors that numerical accuracy can degrade to the point where results cease to be meaningful. Typical problems include the loss of positive-definiteness of the covariance matrix resulting from numerical errors such as finite computer word length and cancelation errors due to the subtraction term in the Kalman covariance update. This type of error may be manifested as a covariance matrix computed erroneously as having negative eigenvalues. In a significant class of filtering problems, propagation of the error covariance matrix, by means of the KF equations, results in a matrix which is not positive semidefinite (a theoretical impossibility). This may occur when 1) the covariance matrix is rapidly reduced by processing very accurate measurements, or 2) a linear combination of state vector components is known with great precision, while other combinations are essentially unobservable. The source of trouble in both cases is the numerical computation of ill-conditioned quantities in finite word length. These errors can cause filter divergence or computations that are so erroneous as to be meaningless. In order to overcome these difficulties, square root formulations of the Kalman filtering have been proposed since they improve the numerical conditioning (robustness) of the solution and achieve twice the effective precision given by conventional Kalman algorithms [17, 59, 103]. One class of algorithms that yields satisfactory results uses a square root factorization of the covariance matrix. Square root factorizations generally have less dynamic range associated with their calculations, and are less susceptible to round-off error. Another advantage of the square root representation is that the covariance matrix does not need to be calculated directly. Instead, it is represented in terms of its square root matrices. This factored form cannot represent a matrix with negative eigenvalues. There are some disadvantages to factored forms of the KF. One significant disadvantage is that they normally require more computations, and therefore more computer time, than the conventional KF. The Potter covariance square root, the Carlson covariance square root, and Bierman–Thornton's U-D filter are three of the more common factorization methods for discrete-time Kalman filtering. Of these factored forms, the U-D filter offers a high degree of numerical precision along with a reasonable increase in the required number of mathematical operations. Because of these properties, the U-D filter is a standard for comparison. The U-D filter is a factorization of the covariance matrix in the form UDU^T, where U is an upper triangular unit matrix (it has ones along the main diagonal) and D is a diagonal matrix. The efficient L-D (or U-D) factorization algorithms for IMM and IMMPDAF are presented in [87, 88]. The system dynamics are modeled by the equations (6)–(7); it is assumed that the model jump process is a Markov process with known transition probabilities p_{ij} (see (9)). The new version of the square root PDAF is approximately two times faster than Kenefic's algorithm [59].

1) *Square Root IMM Filter*: The baseline IMM algorithm was presented in Section IIIA. Here, we summarize the square root covariance updating for the IMM filter (one cycle).

Interaction. Starting with L-D factors of the model conditioned covariances of time $k-1$

$$P_j(k-1 \mid k-1) = L_j(k-1 \mid k-1)D_j(k-1 \mid k-1) \times L_j(k-1 \mid k-1)^T \qquad j = 1,\ldots,N_f$$

we compute the L-D factors of the mixed initial covariances for each of the filters. Note that the factor $\mu_{i|j}(k-1 \mid k-1)$ can be absorbed into $D_j(k-1 \mid k-1)$. To compute the factorization of $P_{0j}(k-1 \mid k-1)$ a rank-one correction is applied to factors of each $P_j(k-1 \mid k-1)$ and the resulting sum of N_f L-D factored terms are combined to form a single L-D factored term. The rank-one correction can be done using the generalized L-D rank-one correction algorithm given in [88]. The merging of N_f L-D factored terms can be accomplished by recursively applying the algorithm, which is also given in [88]. The mixed initial condition for each of the model conditioned filters is computed as usual (see Section IIIA).

Filtering. The state estimates and the factored covariances computed in the interaction step form inputs to filter matched to each of the N_f models. The covariance prediction step $P_j(k \mid k-1)$ can be performed using a modified weighted Gram–Schmidt (MWG-S) algorithm [16]. In [87, 88] the authors present an alternate way of implementing the covariance prediction step, when the process noise covariance is time invariant. This results in substantial computational savings when compared with the MWG-S algorithm. The covariance prediction step involves rank-one corrections to the L-D factors. This also can be accomplished by successively using the L-D rank-one correction algorithm. The proposed algorithm is approximately two times cheaper than MWG-S when the process noise covariance is time invariant. However, if the process noise covariance is time varying, then MWG-S algorithm is preferable since the rank-one corrections approach needs the decomposition at every filter iteration. It can lead to numerical instabilities, because the semidefinite Cholesky might not be numerically robust [88]. By using the results of [87, 88], the need to compute the filter gain matrix $W_j(k)$ and the residual covariance $S_j(k)$ is obviated. Given the L-D factors of $P_j(k \mid k-1)$, the L-D factors of $P_j(k \mid k)$ are determined. The output of this stage are the model conditioned estimates (see Section IIIA (Filtering)), L-D factored covariances for time k [87, 88], likelihood functions

$$\Lambda_j(k) = \frac{1}{(2\pi)^{m'/2}[\prod_{i=1}^{m'} a_{ij}]^{1/2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{m'} \frac{(\rho_{ij})^2}{a_{ij}}\right)$$

where m' is the number of measurements originated from the object in track at time k, $\{\rho_{ij}\}$ and $\{a_{ij}\}$ are the sequential innovation and sequential innovation covariance of the ith scalar component of measurement vector for the filter matched to the jth model (these are computed during the covariance, update recursion of the filter matched to the jth model). The likelihoods are used to update the model probabilities (see Section IIIA (Filtering)).

Combination. The overall state estimate is computed as usual. The covariance mixture equation is very similar in structure to the interaction mixture equation of Interaction mixing step. Therefore, the strategy followed here is the same. Let

$$P_j(k \mid k) = L_j(k \mid k)D_j(k \mid k)L_j(k \mid k)^T \qquad i = 1,2,\ldots,N_f.$$

Note that the factor $\mu_j(k)$ can be absorbed into $D_j(k \mid k)$. To compute the factorization of $P(k \mid k)$, it is needed to apply a rank-one correction to factors of each of $P_j(k \mid k)$ and the resulting sum of N_f L-D factored terms must be combined to form a single L-D factored term. The rank-one correction can be done using the generalized L-D rank-one correction algorithm also. The merging of N_f factored terms can be accomplished by recursively applying the algorithm which is also given in [88].

2) *Square Root IMMPDAF*: The square root IMMPDAF is as described above for the square root IMM except for the following modifications.

Model Conditioned Filtering. Since the standard KF are replaced by PDAFs, the covariance prediction and update recursions for the square root PDAF, given in [87, 88] are used for the N_f filters. The output of the stage are the model conditioned estimates and L-D factored covariances for time k.

Computation of Likelihoods. The likelihoods can be expressed as

$$\Lambda_j(k) = d(k)\left[b'(k)(2\pi)^{m'/2}\left(\prod_{i=1}^{m'} a_{ij}\right)^{1/2} + \sum_{l=1}^{m'_k} \exp\left(-\frac{1}{2}\sum_{i=1}^{m'} \frac{(\rho_{ijl})^2}{a_{ij}}\right)\right]$$

where

$$b'(k) = m'_k V_k^{-1} \frac{1 - P_D P_G}{P_D P_G}$$

$$d(k) = V_k^{-m'_k+1} \frac{P_D P_G}{m'_k}$$

where V_k is the volume of the common validation region at time k, m'_k is the number of validated measurements at time k, P_D is the probability of detection, P_G is the probability that the target measurement falls in the validation region, $\{\rho_{ijl}\}$ and $\{a_{ij}\}$ are the sequential innovation and its variance corresponding to the ith scalar component of lth validated measurement vector for the filter matched to jth model (these are computed during the covariance update recursion of PDAF matched to jth model).

The rest of the algorithm is the same as that of the square root IMM filter.

IV. COMPARISON OF THE MAIN MM ALGORITHMS

Two maneuvering target tracking techniques were compared in [8]. The first technique, called input estimation (see, e.g., the text [14]), models the maneuver as constant unknown input, estimates its magnitude and onset time and then corrects accordingly the state estimate. The second approach models the maneuver as a switching of the target state model, where the various state models can be of different dimension and driven by process noises of different intensities, and estimates the state and mode via the IMM algorithm. While the input estimation algorithm requires around 20 parallel filters, it was shown that the IMM performs equally or better with 2 or 3 filters.

The paper [5] investigates and compares the performance of the IMM and Viterbi algorithms, applied to the problem of radar tracking of maneuvering target. The problem considers a kinematic model in which the acceleration during a maneuver is almost constant and its mean value is one of a finite set of accelerations. The perturbation upon this constant acceleration is modeled as white zero-mean Gaussian noise. The authors consider two different tracks and twelve models, each being assigned a specific acceleration value ranging from 0 to $4g$ (= 40 m/s^2). In the first track the acceleration level of the target almost matches the acceleration level of one of the system models. In the second track, there is a significant difference between the acceleration of the target and the acceleration level of the closest model. It is shown that the performance depends strongly on the parameter

$$\eta = \frac{|\Delta\bar{a}|(\Delta\tau^2)}{\sigma_r},$$

in which $|\Delta\bar{a}|$ the maximal difference between the acceleration levels of any two models, $\Delta\tau$ is the time interval between measurements and σ_r is the standard deviation of the measurement error. In general, when this parameter is relatively large, both algorithms perform well. When η is relatively small, the Viterbi algorithm is better. However, during the beginning of a track and after the start of a maneuver the IMM algorithm provides better estimates. This is especially important when data association is critical.

Recently the IMM has been tested on real multisensor ATC data in [102]. The IMM estimator was implemented in conjunction with a 2-dimensional assignment algorithm that associated the latest scan measurements to the established tracks using a ML criterion. The performance of the IMM was significantly superior to that of a KF used as comparison in all the following aspects:

1) horizontal prediction error,
2) vertical prediction error,
3) number of measurements correctly assigned to tracks

on an extensive database consisting of over 50 targets from 2 radars. Also, the IMM/assignment algorithm tracked with no difficulty the aircraft from the test flight reported in [70].

V. PARALLEL IMPLEMENTATION OF IMM ALGORITHMS

The IMM algorithm basically consists of a group of filters which run in parallel, and a global computation process which collects the results of the filters and produces the output estimation. However, in real applications, processing time may become impractical. Multiprocessors seem to offer the throughput needed for such applications. In the paper [6] the authors investigate the performance of IMM estimator, when implemented on a general purpose shared-memory and a shared-bus MIMD (Multiple Instruction Multiple Data) multiprocessor.

The implementation of this algorithm on the parallel processor is based on the concept of one processor acting as a manager which performs the global computations, and all other processors acting as workers. Both the manager and the workers are implemented as local tasks. The assignments of the manager during each time slice are: getting new measurements, generating the output based on results computed by the other processors, synchronizing between processors, and updating some variables which are part of the EKF and are relevant to all processors.

Each time slice consists of a large number of computations, most of which are an EKF step performed independently for each possible current model. The input of each such filter is an estimate of the system state vector along with its covariance for the previous time step and a new measurement. The output is an updated estimate. This computational load can be divided among the processors almost without additional overhead by assigning each processor a subgroup of the models.

The computational complexity (both the time and space (memory) complexity) as well as the speed-up and efficiency are examined in detail. The time complexity is evaluated using four categories: elementary arithmetic operations $(+,-,\cdot,/)$, complex arithmetic operations (exp( ), arctan( )), accesses to local memory, and access to shared memory.

The space complexity is evaluated using two categories: shared memory and local memory. The basic unit in calculations is the space needed to store a floating variable which is 4 bytes.

In the study reported by [4] the objective was to implement the IMM algorithm in a parallel structure (on a network of four transputers). The experimental work has been done on transputer development system (TDS) using the Occam programming language. Three different architectures for parallel implementation of the IMM algorithm were examined, each with different task partitions for the manager and the workers. A speedup of approximately 2.7 was achieved for the 13 filter IMM implementation. This was further improved to 3.2 by using a time-slipping calculation method which was shown experimentally to have a negligible effect on the average tracking accuracy.

More recent work on parallelization of an IMM-based large scale MTMST algorithm has been presented in [86]. It was shown in this work that parallelization of an IMM becomes practical only if the number of models is large (see Table I).

Highly efficient parallelization (around 100%) was shown in [86] to be achievable on a large scale MT MST problem with IMM trackers by doing the parallelization *across* the IMMs, rather than within the IMMs.

TABLE I
Efficiency of Parallelization of IMM on 4-CPU SPARCserver 1000 Multiprocessor

# Models in IMM	Speed-up	Efficiency
1	0.67	0.17
2	0.73	0.18
3	0.83	0.21
5	1.17	0.29
9	1.98	0.49
11	2.24	0.56
13	2.43	0.60

VI. NONSIMULATION PERFORMANCE PREDICTION TECHNIQUES FOR IMM

Performance prediction of IMM algorithms only via costly and time-consuming Monte-Carlo simulations does not provide sufficient insight into the problems. Unfortunately, none of the earlier nonsimulation techniques [52, 64, 81, 91] can be applied to hybrid state estimation schemes based on MM, such as the IMM algorithm. Error bounding techniques (see e.g. [47, 50, 60, 97]) can provide at most only bounds. In the paper [66] an effective hybrid approach to the nonsimulation performance prediction of IMM systems was developed. In view of the fact that the performance of a MM scheme is scenario dependent, this approach is based on a stochastic performance measure (i.e., conditional expectation of the error covariance) of hybrid nature in the sense that it is a continuous-valued matrix function of a discrete-valued sequence—the system mode sequence, which constitutes the scenario of the problem of interest. This hybrid approach is obtained by using a conditional expectation operation, through which the randomness of the error covariance due to the uncertainties in the continuous subspace of the hybrid space is averaged out whereas the information of the system mode jumping is retained. To calculate efficiently the performance measure, an off-line recursion has been obtained based on conditional successive one-step-ahead-predictions. The capability of the approach in predicting quantitatively the average performance of the algorithm was illustrated via two important examples: a generic maneuvering target tracking problem in ATC and a nonstationary noise identification problem. The predicted performance given by this approach agrees remarkably well with the simulated one, which verifies the good accuracy of the approach and therefore it can be used as an efficient tool for design, sensitivity analysis, and so on.

The philosophy of this conditional expectation based approach is quite general and can be applied to many other algorithms for hybrid systems. For example, the performance of the STDM-based IMM estimator [37, 39] can be evaluated using this approach with suitable modifications. The approach can be readily extended to the variable structure IMM estimator [65], square root IMM algorithm [87, 88], correlated measurement noise IMM estimator [55], IMBM and IMAM algorithms [21].

VII. USING AN IMM ESTIMATOR FOR A PHASED ARRAY RADAR

The use of the IMM algorithm for computing measurement times for a phased array radar is illustrated by [22, 24] for the benchmark problem [23], where the average sample was about 4 s for an aircraft in a holding pattern and about two seconds for a manned aircraft performing $7g$ maneuvers. A revisit technique that responds to target maneuvers and avoids the decision-directed approach to both tracking and sample rate adjustment can be accomplished using the IMM algorithm. Since the output error covariance of the IMM algorithm reflects better the uncertainty in target motion during a maneuver than that of single model filters, the predicted error covariance of the IMM algorithm can be used to compute the time for the next measurement, so that a given level of the performance is maintained during a maneuver. The output of the IMM algorithm was used to compute the time of the next measurement, or radar dwell, to keep track loss below a given level. By including maneuvering and nonmaneuvering models in the IMM algorithm, the sample times automatically decreased during maneuvers to maintain the desired level of performance. Thus, using the output of the IMM algorithm gives measurement or dwell times that automatically reflect target maneuvers, target range, missed detections, and weak and strong signal returns. Also, the probability of the nonmaneuvering mode within the IMM algorithm was used to adjust the energy of the waveform. By increasing the energy during maneuvers, the accuracy of the state estimates was improved significantly.

Another design and the performance of an IMM estimator for the benchmark problem [23] of highly maneuvering targets are presented in [45]. The design parameters of the IMM algorithm as well as an adaptive sampling policy are described. It is shown that the IMM filter with adaptive sampling policy has a better performance in terms of the average track loss, averaged dwells per run and noise reduction in nonmaneuvering periods of time than the best KF. The design was done based on the first four benchmark trajectories. The evaluations were done for these four plus two additional trajectories. In [45] the authors considered three modes: the first mode is a nearly constant velocity motion (second order per coordinate with a low level white noise acceleration). In the second mode, the target dynamic model has a relatively higher level of noise. This mode corresponds to an on-going maneuver. Finally, the third mode of operation, which is a Wiener process acceleration model, corresponds to the situations that the target starts/terminates a maneuver (large acceleration increments). The tracking algorithm is an IMM estimator with three models corresponding to the above three modes of the dynamic system. The evolution of the hybrid system among the modes is modeled as a first order Markov chain. The covariance of the noise, contributing to the evolution of the state in each mode, is determined based on the maximum acceleration corresponding to that mode. The tracking algorithm also selects the revisit (dwell) time for the target. The interval to the next dwell is selected from a predetermined set of values as the largest interval, such that the prediction of the measurement errors are smaller than a certain selected threshold related to the radar's beamwidth. Beam pointing control is done based on the prediction of the target coordinates at the selected revisit time and pointing the beam to this direction. The range gate of the radar is centered at the predicted range of the target at the revisit time. The pointing accuracy (how much the target is off the beam center) is accounted for in the next measurement since it affects the SNR and the measurement accuracies depend on the SNR.

The probability transition matrix of the Markov chain is designed based on the expected sojourn time in each mode of the hybrid system [14]. In [45] a predetermined set of sampling intervals is used, with the selection done based on the prediction of the radar angle innovation standard deviations compared with the radar beamwidth. At each time step, the estimator predicts the innovation covariance for the largest possible sampling interval. The uncertainty, caused by the fluctuations of the target cross-section, is also considered in the calculations of the variances. Then, the standard deviations of both angle innovations are compared with a fraction of the antenna beamwidth. If any of the azimuthal or elevation angle innovation standard deviations exceeds this threshold, the candidate sampling interval is rejected and the test is repeated for the next smaller sampling interval. Sampling for initialization is done with the maximum allowable rate (10 Hz). This rate lasts for a total of four steps to assure a fast decrease of the initial errors. After the sampling interval is determined, the position of the target is predicted and used to point the beam.

The simulation results of the work reveal that the IMM estimator in conjunction with the adaptive sampling policy yields satisfactory results in terms of track loss percentage and average dwells per run. This comes from the maneuver adaptability of the IMM algorithm. A single model KF with the same adaptive sampling policy does not yield satisfactory results. Even in comparison with the KF with 1 s constant sampling, the IMM with the average sampling interval of about 2 s has less track loss. Another advantage of the IMM algorithm is the noise reduction in the state estimate, especially, in the velocity components of the state.

The second, more realistic, benchmark problem [26] included false alarms and ECM (electronic countermeasures), specifically RGPO (range gate pull-off) and SOJ (stand-off jammer). The solutions for this problem presented in [61, 98] used the IMMPDAF while the one of [20] used the MHT and an IMM. The latter also proposed the combination of the MHT with the IMM, but this was not implemented.

VIII. IMM ALGORITHMS FOR TARGET TRACKING WITH GLINT NOISE AND NOISE IDENTIFICATION

The optimality of the KF is based on the assumption of having zero-mean white Gaussian noises. If this assumption is violated, the KF is no longer the optimal filter. A non-Gaussian noise arising in a radar system is known as a glint noise. The glint noise distribution is long tailed and seriously affects the tracking performance. Filtering in non-Gaussian environments has been studied by many researchers. Robust preprocessing methods for KF have been proposed in [57, 75]. In the method of [74] a nonlinear score function was introduced which yields a corrective term in the state estimate equation, while retaining the computationally appealing recursive structure of the KF. In this method it is assumed that either the process noise or the measurement noise is non-Gaussian. However, since it requires a numerical convolution operation to evaluate the score function , this limits the practicality of this method.

New algorithms were developed by Wu [101]. These algorithms incorporate the nonlinear Masreliez filter [75] into the IMM method. They substitute the KF in the IMM algorithm with the nonlinear Masreliez filter and modify the estimate of model probabilities. Simulations show that these new algorithms not only can filter the glint noise efficiently, but also respond quickly to maneuvering. Although Wu has used the score function method with combination of IMM approach, he has avoided the convolution operation and simplified the evaluation of the score function by applying a normal expansion for the distribution of the measurement prediction. However, the approximate spherical model [79] used to decouple the state components is not a good approximation [46].

The application of the IMM approach to the target tracking problem, when the measurements are perturbed by glint noise, is considered also in [46]. A good approximation for a glint noise with heavy tails is a mixture of a Gaussian noise with a moderate variance and a Laplacian distributed noise, with high variance and low occurrence probability. Therefore, for this problem the authors considered an IMM algorithm with two filters. One filter is matched to the dynamic system with Gaussian measurement noise and the other is matched to the same dynamic system but with a high variance Laplacian distributed noise. With target motion described in Cartesian coordinates and measurements in the radar coordinates, the former filter is an EKF. Even though one can obtain a nonlinear filter that accounts explicitly for the Laplacian distributed measurement noise, it is expensive due to the need to perform numerical integration in real time. The results of this work show that a KF with moment-matched measurement noise variance, performs close to the optimum. Therefore, the filter matched to the second model is also an EKF. The authors showed that this method performs better than the score function method and it is also robust to the variations of the glint noise parameters.

In [68] an approach to noise identification is proposed which uses hybrid system models with a suitable estimation scheme. This approach is valid for non-stationary noise with rapidly or slowly varying statistics as well as stationary noise. The IMM algorithm was used in the approach. The proposed approach is powerful in the sense that it can be used to identify simultaneously various noise parameters in linear and nonlinear systems. It is also flexible in terms of model design.

IX. CONCLUSION

While no tracking algorithms can work in an arbitrarily dense multitarget-multisensor environment, the recently developed IMM techniques described in this paper extend the range in which one can achieve reliable tracking performance, i.e., small track loss percentage. Compromises between computational requirements and performance will be necessary in many cases, especially where the data rate is high and the available computing power is limited, as in airborne systems. The assumption made in the various algorithms discussed here will probably be modified according to the needs of the specific applications.

A number of problems that are yet open can be seen at this time. One such problem is the treatment of multisensor group tracking of maneuvering targets in clutter [18]. At this point it is not known whether filtering techniques can be combined with any of the methods cited above. The application of new techniques to phased-array radars, with special consideration to the necessary scanning patterns, should also be continued [21, 23, 36, 45, 95]. The performance of the IMM estimator can be further improved by fine-tuning of the design parameters.

The IMM procedure has been established on a solid theoretical basis and proved to be quite appropriate for the maneuvering target tracking problem [94]. The investigations documented in the literature indicate that the IMM algorithm is the superior technique for multisensor-multitarget tracking and data association.

The recent activity in the area of IMM methods indicates that this emerging body of knowledge experiences a rapid transition to becoming relevant to engineering practice by solving real problems.

ACKNOWLEDGMENTS

E. Mazor wishes to thank S. Rogers, S. Levin and B. Golubev from Elta Corp., Israel, for introducing the problem and continuing support through all the stages of the work.

REFERENCES

[1] Ackerson, G. A., and Fu, K. S. (1970)
On state estimation in switching environments.
IEEE Transactions on Automatic Control (Feb. 1970), 10–17.

[2] Alouani, A. T., and Blair, W. D. (1991)
Use of a kinematic constraint in tracking constant speed, maneuvering targets.
In *Proceedings of the 30th Conference on Decision and Control*, Brighton, England, Dec. 1991, 2055–1058.

[3] Alouani, A. T., Xia, P., Rice, T. R., and Blair, W. D. (1991)
A two-stage Kalman estimator for state estimation in the presence of random bias and for tracking manuevering targets.
In *Proceedings of the 30th IEEE Conference on Decision and Control*, Brighton, England, Dec. 1991, 1059–1062.

[4] Atherton, D. P., and Lin, H.-J. (1994)
Parallel implementation of IMM tracking algorithm using transputers.
IEE Proceedings on Radar Sonar Navigation, **141**, 6 (Dec. 1994), 325–332.

[5] Averbuch, A., Itzikowitz, S., and Kapon, T. (1991)
Radar target tracking—Viterbi versus IMM.
IEEE Transactions on Aerospace and Electronic Systems, **27**, 3 (May 1991), 550–563.

[6] Averbuch, A., Itzikowitz, S., and Kapon, T. (1991)
Parallel implementation of multiple model tracking algorithms.
IEEE Transactions on Parallel and Distributed Systems, **2**, 2 (Apr. 1991), 242–251.

[7] Bar-Shalom, Y., and Fortmann, T. E. (1988)
Tracking and data association.
New York: Academic Press, 1988.

[8] Bar-Shalom, Y., Chang, K. C., and Blom, H. A. P. (1989)
Tracking a maneuvering target using input estimation versus the interacting multiple model algorithm.
IEEE Transactions on Aerospace and Electronic Systems, **25**, 2 (Mar. 1989), 296–300.

[9] Bar-Shalom, Y. (1989)
Recursive tracking algorithms: From the Kalman filter to intelligent trackers for cluttered environment.
In *Proceedings of the 1989 IEEE International Conference on Control and Applications*, Jerusalem, 1989.

[10] Bar-Shalom, Y. (Ed.) (1990)
Multitarget-multisensor tracking: Advanced applications.
Norwood, MA: Artech House, 1990.

[11] Bar-Shalom, Y., Chang, K. C., and Blom, H. A. P. (1990)
Automatic track formation in clutter with a recursive algorithm.
In Y. Bar-Shalom (Ed.), *Multitarget-Multisensor Tracking: Advanced Applications.*
Norwood, MA: Artech House, 1990, 25–42.

[12] Bar-Shalom, Y. (Ed.) (1992)
Multitarget-Multisensor Tracking: Applications and Advances, Vol. II.
Norwood, MA: Artech House, 1992.

[13] Bar-Shalom, Y., Chang, K. C., and Blom, H. A. P. (1992)
Tracking splitting targets in clutter by using an interacting multiple model joint probabilistic data association filter.
In Y. Bar-Shalom (Ed.), *Multitarget-Multisensor Tracking: Applications and Advances*, Vol. II.
Norwood, MA: Artech House, 1992, 93–110.

[14] Bar-Shalom, Y., and Li, X. (1993)
Estimation and tracking: Principles, techniques, and software.
Norwood, MA: Artech House, 1993.

[15] Bar-Shalom, Y., and Li, X. R. (1995)
Multitarget-Multisensor Tracking: Principles and Techniques.
Storrs, CT: YBS Publishing, 1995.

[16] Bierman, G. J., and Thornton, C. L. (1977)
Numerical comparison of Kalman filter algorithms: Orbit determination case study.
Automatica, **13** (Jan. 1977), 23–27.

[17] Bierman, G. J. (1977)
Factorization Methods for Discrete Sequential Estimation.
New York: Academic Press, 1977.

[18] Blackman, S. S. (1986)
Multiple Target Tracking with Radar Applications.
Norwood, MA: Artech House, 1986.

[19] Blackman, S. S. (1986)
Multitarget tracking with an agile beam radar.
In Y. Bar-Shalom (Ed.), *Multitarget-Multisensor Tracking: Applications and Advances*, Vol. II.
Norwood, MA: Artech House, 1986, 237–269.

[20] Blackman, S. S., Busch, M. T., and Popoli, R. F. (1995)
IMM/MHT tracking and data association for benchmark tracking problem.
In *Proceedings of the American Control Conference*, Seattle, WA, June 1995, 2606–2610.

[21] Blair, W. D., and Watson, G. A. (1992)
Interacting multiple bias model algorithm with application to tracking maneuvering targets.
In *Proceedings of the 31st Conference on Decision and Control*, Tucson, AZ, Dec. 1992, 3790–3795.

[22] Blair, W. D. (1994)
Toward the integration of tracking and signal processing for phased array radar, signal and data processing for small targets.
Orlando, FL, Apr. 1994.

[23] Blair, W. D., Watson, G. A., and Hoffman, S. A. (1994)
Benchmark problem for beam pointing control of phased array radar against maneuvering targets.
In *Proceedings of the 1994 American Control Conference*, Baltimore, MD, June 1994.

[24] Blair, W. D., and Watson, G. A. (1994)
IMM algorithm for solution to benchmark problem for tracking maneuvering targets.
In *Proceedings of Acquisition, Tracking, and Pointing Conference*, SPIE Orlando '94, Apr. 1994.

[25] Blair, W. D., and Kazakos, D. (1994)
Estimation and detection for systems with second-order Markovian switching coefficients.
In *Proceedings of 1994 American Control Conference*, Baltimore, MD, June 1994.

[26] Blair, W. D., Watson, G. A., Gentry, G. L., and Hoffman, S. A. (1995)
Benchmark problem for beam pointing control of phased array radar against maneuvering targets in the presence of ECM and FA.

In *Proceedings of the American Control Conference*, Seattle, WA, June 1995, 2601–2605.

[27] Blair, W. D., and Bar-Shalom, Y. (1996)
Tracking maneuvering targets with multiple sensors: Does more data always mean better estimates?
IEEE Transactions on Aerospace and Electronic Systems, **32**, 1 (Jan. 1996), 450–455.

[28] Blom, H. A. P. (1984)
An efficient filter for abruptly changing systems.
In *Proceedings of the 23rd IEEE Conference on Decision and Control*, Las Vegas, NV, Dec. 1984, 656–658.

[29] Blom, H. A. P. (1985)
An efficient decision-making-free filter for processes with abrupt changes.
In *Proceedings of the IFAC Symposium on Identification and System Parameter Estimation*, 1985, 631–636.

[30] Blom, H. A. P., and Bar-Shalom, Y. (1988)
The interacting multiple model algorithm for systems with Markovian switching coefficients.
IEEE Transactions on Automatic Control, **33**, 8 (Aug. 1988), 780–783.

[31] Blom, H. A. P., and Bar-Shalom, Y. (1990)
Time-reversion of a hybrid state stochastic difference system with a jump-linear smoothing application.
IEEE Transactions on Information Theory, **36**, 4 (July 1990), 836–847.

[32] Blom, H. A. P. (1991)
Hybrid state estimation for systems with semi-Markov switching coefficients.
In *Proceedings of the European Control Conference*, Grenoble, July 1991.

[33] Blom, H. A. P., Hogendoorn, R. A., and Van Doorn, B. A. (1992)
Design of a multisensor tracking system for advanced air traffic control.
In *Multitarget-Multisensor Tracking: Applications and Advances*, Vol. II.
Norwood, MA: Artech House, 1992, 31–63.

[34] Bogler, P. L. (1987)
Tracking a maneuvering target using input estimation.
IEEE Transactions on Aerospace and Electronic Systems, **AES-23**, 3 (May 1987), 298–310.

[35] Bogler, P. L. (1990)
Radar Principles with Applications to Tracking Systems.
New York: Wiley, 1990.

[36] Brookner, E. (Ed.) (1991)
Practical Phased Array Antenna Systems.
Boston: Artech House, 1991.

[37] Campo, L., Mookerjee, P., and Bar-Shalom, Y. (1988)
Failure detection via recursive estimation for a class of semi-Markov switching systems.
In *Proceedings of the 27th IEEE Conference on Decision and Control*, Austin, TX, Dec. 1988, 1966–1971.

[38] Campo, L., and Bar-Shalom, Y. (1990)
Control of discrete-time hybrid stochastic systems.
In *Proceedings of the 1990 American Control Conference*, Boston, June 1990, 7–12.

[39] Campo, L., Mookerjee, P., and Bar-Shalom, Y. (1991)
State estimation for systems with sojourn-time-dependent Markov model switching.
IEEE Transactions on Automatic Control, **36**, 2 (Feb. 1991), 238–243.

[40] Chang, C. B., and Athans, M. (1978)
State estimation for discrete system with switching parameters.
IEEE Transactions on Aerospace and Electronic Systems, **AES-14** (May 1978), 418–425.

[41] Chang, K. C., Chong, C. Y., and Bar-Shalom, Y. (1986)
Joint probabilistic data association in distributed sensor networks.
IEEE Transactions on Automatic Control, **31**, 10 (Oct. 1986), 889–897.

[42] Chang, K. C., and Bar-Shalom, Y. (1987)
Distributed multiple model estimation.
In *Proceedings of the American Control Conference*, 1987, 797–802.

[43] Chang, K. C., and Bar-Shalom, Y. (1989)
Distributed adaptive estimation with probabilistic data association.
Automatica, **25**, 3 (1989), 359–369.

[44] Chong, C. H., Mori, S., and Chang, K. C. (1985)
Information fusion in distributed sensor networks.
Proceedings of the American Control Conference (1985), 830–835.

[45] Daeipour, E., Bar-Shalom, Y., and Li, X. (1994)
Adaptive beam pointing control of a phased array radar using an IMM estimator.
In *Proceedings of the 1994 American Control Conference*, Baltimore, MD, June 1994.

[46] Daeipour, E., and Bar-Shalom, Y. (1995)
An interacting multiple model approach for target tracking with glint noise.
IEEE Transactions on Aerospace and Electronic Systems, **31**, 2 (Apr. 1995).

[47] Daum, F. E. (1990)
Bounds on performance for multiple target tracking.
IEEE Transactions on Automatic Control, **35**, 4 (Apr. 1990), 443–446.

[48] Dufour, F., and Mariton, M. (1991)
Tracking a 3D maneuvering target with passive sensors.
IEEE Transactions on Aerospace and Electronic Systems, **27**, 4 (July 1991), 725–738.

[49] Dufour, F., and Mariton, M. (1992)
Passive sensor data fusion and maneuvering target tracking.
In Y. Bar-Shalom (Ed.), *Multitarget-Multisensor Tracking: Applications and Advances*, Vol. II.
Norwood, MA: Artech House, 1992, 65–92.

[50] Ezzine, J., and Haddad, A. H. (1988)
Error bounds in the averaging of hybrid systems.
In *Proceedings of the 27th IEEE Conference on Decision and Control*, Austin, TX, Dec. 1988, 1787–1791.

[51] Farina, A., and Pardini, S. (1980)
Survey of radar data-processing techniques in air-traffic-control and surveillance systems.
IEE Proceedings, **127**, Pt. F, 3 (June 1980), 190–204.

[52] Fortmann, T. E., Bar-Shalom, Y., Scheffe, M., and Gelfand, S. (1985)
Detection thresholds for tracking in clutter—A connection between estimation and signal processing.
IEEE Transactions on Automatic Control, **AC-30**, 5 (Mar. 1985), 221–229.

[53] Friedland, B. (1969)
Treatment of bias in recursive filtering.
IEEE Transactions on Automatic Control, **AC-14**, 4 (Aug. 1969), 359–367.

[54] Gauvrit, M. (1984)
Bayesian adaptive filter for tracking with measurements of uncertain origin.
Automatica, **20** (1984), 217–224.

[55] Guu, J. A., and Wei, C. H. (1991)
Maneuvering target tracking using IMM method at high measurement frequency.
IEEE Transactions on Aerospace and Electronic Systems, **27**, 3 (May 1991), 514–519.

[56] Helmick, R. E., Blair, W. D., and Hoffman, S. A. (1995)
Interacting multiple model approach to fixed-interval smoothing.
IEEE Transactions on Information Theory (Nov. 1995).

[57] Hewer, G. A., Martin, R. D., and Seh, J. (1987)
Robust preprocessing for Kalman filtering of glint noise.
IEEE Transactions on Aerospace and Electronic Systems, **AES-23**, 1 (Jan. 1987), 120–128.

[58] Houles, A., and Bar-Shalom, Y. (1989)
Multisensor tracking of a maneuvering target in clutter.
IEEE Transactions on Aerospace and Electronic Systems, **25**, 2 (Mar. 1989), 176–189.

[59] Kenefic, R. (1990)
Active sonar application of a U-D square root PDAF.
IEEE Transactions on Aerospace and Electronic Systems, **26**, 5 (Sept. 1990), 850–857.

[60] Kerr, T. H. (1989)
Status of CR-like lower bounds for nonlinear filtering.
IEEE Transactions on Aerospace and Electronic Systems, **25**, 5 (Sept. 1989), 590–601.

[61] Kirubarajan, T., and Bar-Shalom, Y. (1995)
Adaptive beam pointing control of a phased array radar in the presence of ECM and FA using IMMPDAF.
In *Proceedings of the American Control Conference*, Seattle, WA, June 1995, 2616–2620.

[62] Leondes, C. T., and Pearson, J. O. (1973)
Kalman filtering of systems with parameter uncertainties—A survey.
International Journal of Control, **17**, 4 (1973), 785–801.

[63] Lerro, D., and Bar-Shalom, Y. (1993)
Interacting multiple model tracking with target amplitude feature.
IEEE Transactions on Aerospace and Electronic Systems, **29**, 2 (Apr. 1993), 494–508.

[64] Li, X. R., and Bar-Shalom, Y. (1991)
Stability evaluation and track life of PDAF for tracking in clutter.
IEEE Transactions on Automatic Control, **36**, 5 (May 1991), 588–602.

[65] Li, X. R., and Bar-Shalom, Y. (1992)
Mode-set adaptation in multiple-model estimators for hybrid systems.
In *Proceedings of the 1992 American Control Conference*, Chicago, IL, June 1992, 1794–1799.

[66] Li, X. R., and Bar-Shalom, Y. (1992)
Performance prediction of the interacting multiple model algorithm.
In *Proceedings of the 1992 American Control Conference*, Chicago, IL, June 1992, 2109–2113; also in *IEEE Transactions on Aerospace and Electronic Systems*, **29**, 3 (July 1993), 755–771.

[67] Li, X. R., and Bar-Shalom, Y. (1993)
Design of an interacting multiple model algorithm for air traffic control tracking.
IEEE Transactions on Control Systems Technology, **1**, 3 (Sept. 1993), 186–194.

[68] Li, X. R., and Bar-Shalom, Y. (1994)
A recursive multiple model approach to noise identification.
IEEE Transactions on Aerospace and Electronic Systems, **30**, 3 (July 1994), 671–684.

[69] Li, X. R., and Bar-Shalom, Y. (1996)
Multiple-model esitamtion with variable structure.
IEEE Transactions on Automatic Control, **41**, 4 (Apr. 1996), 1–16.

[70] Liggins, M. E., et al. (1991)
Multispectral fusion for low observable surveillance: A system perspective.
In *Proceedings of the 1991 Joint Service Data Fusion Symposium*, Oct. 1991.

[71] Lin, J., and Atherton, D. P. (1993)
An investigation of the SFIMM algorithm for tracking maneuvering targets.
In *Proceedings of the 32nd Conference on Decision and Control*, San Antonio, TX, Dec. 1993, 930–935.

[72] Lu, M., Qiao, X., and Chen, G. (1992)
A parallel square-root algorithm for modified extended Kalman filter.
IEEE Transactions on Aerospace and Electronic Systems, **28**, 1 (Jan. 1992), 153–163.

[73] Mariton, M., and Sworder, D. (1992)
Maneuvering target tracking: Imaging and non-imaging sensors.
In C. T. Leondes (Ed.), *Control and Dynamic Systems: Advances in Theory and Applications*.
New York: Academic Press, 1992, 483–516.

[74] Masreliez, C. J. (1975)
Approximate non-Gaussian filtering with linear state and observation relations.
IEEE Transactions on Automatic Control (Feb. 1975), 107–110.

[75] Masreliez, C. J., and Martin, R. D. (1977)
Robust Bayesian estimation for the linear model and robustifying the Kalman filter.
IEEE Transactions on Automatic Control (1977), 361–371.

[76] Maybeck, P. S., and Schore, M. R. (1990)
Robustness of a moving-bank multiple model adaptive algorithm for control of a flexible spacestructure.
In *Proceedings of the National Aerospace and Electronics Conference*, NAECON 90, Dayton, 1990, 368–374.

[77] Millnert, M. (1980)
An approach to recursive identification of abruptly changing systems.
In *Proceedings of the 19th IEEE Conference on Decision and Control* (1980), 1026.

[78] Mookerjee, P., Campo, L., and Bar-Shalom, Y. (1987)
Estimation in systems with a semi-Markov switching model.
In *Proceedings of the 26th Conference on Decision and Control*, Los Angeles, Dec. 1987, 332–334.

[79] Moose, R. L. (1975)
An adaptive state estimation solution to the maneuvering target problem.
IEEE Transactions on Automatic Control, **AC-20** (1975), 359.

[80] Moose, R. L., Vanlandingham, H. F., and McCabe, D. H. (1979)
Modeling and estimation for tracking maneuvering targets.
IEEE Transactions on Aerospace and Electronic Systems **AES-15**, 3 (May 1979), 448–455.

[81] Mori, S., Chang, K. C., and Chong, C. Y. (1992)
Performance analysis of optimal data association—With applications to multiple target tracking.
In Y. Bar-Shalom (Ed.), *Multitarget-Multisensor Tracking: Applications and Advances*, Vol. II.
Norwood, MA: Artech House, 1992, 183–235.

[82] Munir, A., and Atherton, D. P. (1994)
Maneuvering target tracking using an adaptive interacting multiple model algorithm.
In *Proceedings of the 1994 American Control Conference*, Baltimore, MD, June 1994, 1324–1328.

[83] Nagy, P., and Bier, S. (1990)
Intelligent internetted sensor management systems for tactical aircraft.
In *Proceedings of the National Aerospace and Electronics Conference*, NAECON 90, Dayton, 1990, 321–327.

[84] Pao, L., and Frei, C. W. (1995)
A comparison of parallel and sequential implementations of a multisensor multitarget tracking algorithm.
In *Proceedings of the American Control Conference*, Seattle, WA, June 1995, 2601–2605.

[85] Pattipati, K. R., and Sandell, N. R. (1983)
A unified view of state estimation in switching environments.
In *Proceedings of the 1983 American Control Conference*, 1983, 458–465.

[86] Popp, R., Pattipati, K. R., and Bar-Shalom, Y. (1995)
The parallelization of a large-scale IMM-based multitarget tracking algorithm.
In *Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets*, July 1995.

[87] Raghavan, V., Pattipati, K. R., and Bar-Shalom, Y. (1992)
Efficient square-root algorithms for PDA, IMM and IMMPDA filters.
In *Proceedings of the 31st Conference on Decision and Control*, Tucson, AZ, Dec. 1992, 3680–3685.

[88] Raghavan, V., Pattipati, K. R., and Bar-Shalom, Y. (1993)
Efficient L-D factorization algorithms for PDA, IMM, and IMMPDA filters.
IEEE Transactions on Aerospace and Electronic Systems, **29**, 4 (Oct. 1993), 1297–1310.

[89] Raisbeck, G. (1983)
Viewing the issue.
IEEE Journal of Oceanic Engineering (Special Issue on Positioning, Localization and Tracking), **OE-8**, 3 (July 1983), 110–112.

[90] Rogers, S. R. (1987)
Alpha-beta filter with correlated measurement noise.
IEEE Transactions on Aerospace and Electronic Systems, **AES-23**, 4 (July 1, 1987), 592–594.

[91] Rogers, S. R. (1991)
Diffusion analysis of track loss in clutter.
IEEE Transactions on Aerospace and Electronic Systems, **2**, 2 (Mar. 1991), 380–387.

[92] Sung, T. Y., and Hu, Y. H. (1987)
Parallel VLSI implementation of the Kalman filter.
IEEE Transactions on Aerospace and Electronic Systems, **AES-23**, 2 (Mar. 1987), 215–224.

[93] Tugnait, J. K. (1982)
Detection and estimation for abruptly changing systems.
Automatica, **18**, 5 (1982), 607–615.

[94] Vacher, P., Barret, I., and Gauvrit, M. (1992)
Design of a tracking algorithm for an advanced ATC system.
In Y. Bar-Shalom (Ed.), *Multitarget-Multisensor Tracking: Applications and Advances*, Vol. II.
Norwood, MA: Artech House, 1992, 1–29.

[95] VanKeuk, G., and Blackman, S. S. (1993)
On phased-array radar tracking and parameter control.
IEEE Transactions on Aerospace and Electronic Systems, **29**, 1 (Jan. 1993), 186–194.

[96] Waltz, E., and Llinas, J. (1990)
Multisensor Data Fusion.
Norwood, MA: Artech House, 1990.

[97] Wahsburn, R. B., Allen, T. G., and Teneketzis, D. (1985)
Performance analysis for hybrid state estimation problems.
In *Proceedings of the 1985 American Control Conference*, Boston, MA, June 1985, 1047–1053.

[98] Watson, G. A., and Blair, W. D. (1995)
Solution to second benchmark problem for tracking maneuvering targets in the presence of FA and ECM.
In *Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets*, July 1995.

[99] Watson, G. A., and Blair, W. D. (1995)
Interacting acceleration compensation algorithm for tracking maneuvering targets.
IEEE Transactions on Aerospace and Electronic Systems, **31**, 3 (July 1995).

[100] Willner, D., Chang, C. B., and Dunn, K. P. (1976)
Kalman filter algorithms for a multi-sensor system.
In *Proceedings of the 15th IEEE Conference on Decision and Control* (Dec. 1976), 570–574.

[101] Wu, W. R., and Cheng, P. P. (1994)
A nonlinear IMM algorithm for maneuvering target tracking.
IEEE Transactions on Aerospace and Electronic Systems, **30**, 3 (July 1994), 875–886.

[102] Yeddanapudi, M., Bar-Shalom, Y., and Pattipati, K. (1995)
IMM estimation for multitarget-multisensor air traffic surveillance.
Proceedings of IEEE, **85**, 1 (Jan. 1997), 80–94.

[103] Zeitz, F. H., and Maybeck, P. S. (1993)
An alternate algorithm for discrete-time filtering.
IEEE Transactions on Aerospace and Electronic Systems, **29**, 4 (Oct. 1993), 1123–1136.

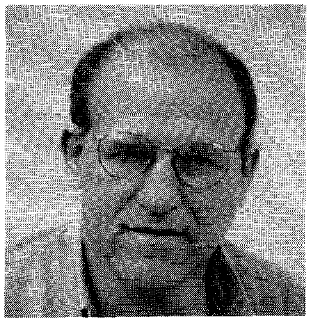
Efim Mazor was born in Kazan, Russia. He received his M.S. and Ph.D. degrees in computer science from Kazan Institute of Aeronautics in 1962 and 1973, respectively.

From 1962 to 1969 he was employed by the Computer Devices Plant and by the Electromechanical Plant, in Kazan, where he was engaged in radar systems work. From 1969 to 1971 he was a postgraduate student and a scientist at the R&D Institute of Radio-Electronics, Moscow. He served as an Associate Professor and Group Leader at the Kazan Institute of Aeronautics. In 1991 Dr. Mazor immigrated to Israel and joined the Technion–Israel Institute of Technology in Haifa, where he pursued his interests in the development of efficient algorithms for multisensor target tracking, aircraft group recognition, image processing and machine vision. Since 1996 he has been working for Kulicke & Soffa Co., Israel, in their Vision Engineering Group. His research interests include estimation theory, target tracking, data association and micro-processor based systems.

Joshua Dayan received his B.Sc. and M.Sc. degrees in chemical engineering from Technion–Israel Institute of Technology, in 1961 and 1964, respectively, and his Ph.D. from Illinois Institute of Technology, Chicago, in 1967.

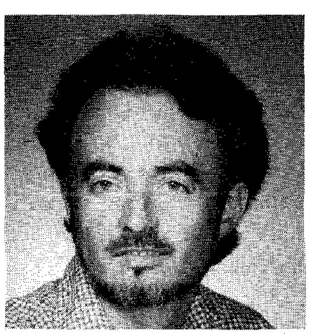
After completing his studies he spent several years with Amoco, near Chicago and with The Industrial Automation Institute in Israel, working in the area of computer control of industrial processes. Since 1971 he has been on the Faculty of Mechanical Engineering at Technion in Haifa. His teaching and research are in control applications and robotics. He is extensively involved in industrial consulting work both in Israel and the United States. His interests and research areas are in power generation systems, process control and robotics.

Prof. Dayan had supervised the research work of over 30 graduate students and published over 80 papers in journals and in conference proceedings.



Amir Averbuch was born in Tel Aviv, Israel. He got his B.Sc. and M.Sc. degrees in mathematics from the Hebrew University in Jerusalem, in 1971 and 1975, respectively. He received his Ph.D. degree in computer science from Columbia University, New York, in 1983.

During 1976–1986 he was a research staff member in IBM T. J. Watson Research Center, Dept. of Computer Science. Since October 1987, he has been an Assistant Professor in the Department of Computer Science, School of Mathematical Sciences, Tel Aviv University. His research interests include parallel processing and wavelet for signal/image processing and numerical computation.



Yaakov Bar-Shalom (S'63—M'66—SM'80—F'84) was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967, and the Ph.D. degree from Princeton University, Princeton, NJ, in 1970, all in electrical engineering.

From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, CA. Currently he is Professor of Electrical and Systems Engineering and Director of the ESP Lab (Estimation and Signal Processing) at the University of Connecticut. His research interests are in estimation theory and stochastic adaptive control and he has published over 200 papers in these areas.

In view of the causality principle between the name of a person (in this case, "(he) will track", in the modern version of the original language of the Bible) and the profession of this person, his interests have focused on tracking. He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate text *Estimation and Tracking: Principles, Techniques and Software* (Artech House, 1993), the text *Multitarget-Multisensor Tracking: Principles and Techniques* (YBS Publishing, 1995), and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Artech House, Vol. I 1990; Vol. II 1992). He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He has been consulting to numerous companies, and originated the series of *Multitarget-Multisensor Tracking* short courses offered via UCLA Extension, at Government Laboratories, private companies, and overseas. He has also developed the commercially available interactive software packages MULTIDAT™ for automatic track formation and tracking of maneuvering or splitting targets in clutter, PASSDAT™ for data association from multiple passive sensors, BEARDAT™ for target localization from bearing and frequency measurements in clutter, IMDAT™ for image segmentation and target centroid tracking and FUSEDAT™ for fusion of possibly heterogeneous multisensor data for tracking. During 1976 and 1977 he served as Associate Editor of the *IEEE Transactions on Automatic Control* and from 1978 to 1981 as Associate Editor of *Automatica.* He was Program Chairman of the 1982 American Control Conference, General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–1987 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–1989 was a member of the Board of Governors of the IEEE CSS. In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 has been a Distinguished Lecturer of the IEEE AESS. He is co-recipient of the M. Barry Carlton Award for the best paper in the *IEEE Transactions on Aerospace and Electronic Systems* in 1996.