Multiple-Model Estimation with Variable Structure Part V: Likely-Model Set Algorithm

X. RONG LI, Senior Member, IEEE
University of New Orleans

YOUMIN ZHANG, Member, IEEE
The University of Western Ontario

A multiple-model (MM) estimator with a variable structure, called likely-model set (LMS) algorithm, is presented, which is generally applicable to most hybrid estimation problems and is easily implementable. It uses a set of models that are not unlikely to match the system mode in effect at any given time. Different versions of the algorithm are discussed. The model set is made adaptive in the simplest version by deleting all unlikely models and activating all models to which a principal model may jump so as to anticipate the possible system mode transitions. The generality, simplicity, and ease in the design and implementation of the LMS estimator are illustrated via an example of tracking a maneuvering target and an example of fault detection and identification. Comparison of its cost-effectiveness with other fixed- and variable-structure MM estimators is given.

Manuscript received September 1, 1998; revised May 14, 1999; released for publication November 10, 1999.

IEEE Log No. T-AES/36/2/05220.

Refereeing of this contribution was handled by W. D. Blair.

This research was supported in part by ONR via Grant N00014-97-1-0570, NSF via Grant ECS-9734285, and LEQSF via Grant (1996–99)-RD-A-32.

Authors' addresses: X. R. Li, Dept. of Electrical Engineering, University of New Orleans, New Orleans, LA 70148, E-mail: (xli@uno.edu); Y. Zhang, Dept. of Electrical and Computer Engineering, The University of Western Ontario, London, Ontario, N6A 589 Canada.

I. INTRODUCTION

Multiple-model (MM) estimation, as a powerful approach to adaptive estimation, has received a great deal of attention in recent years due to its unique power to handle problems with both structural and parametric uncertainties and/or changes, and to decompose a complex problem into simpler subproblems. In his Bode Lecture given at the 1995 IEEE Conference on Decision and Control, Professor Kumpati S. Narendra advocated this powerful adaptive method.

In the MM estimation, it is assumed that the possible system behavior patterns or structures, called system modes, can be represented by a set of models; a bank of filters runs in parallel at every time, each based on a particular model, to obtain the model-conditional estimates; and the overall state estimate is a certain combination of these model-conditional estimates.

Initiated by Magill [32], the early work on MM estimation considered only systems with a time-invariant unknown or uncertain system mode. Many applications (or reinventions) of this MM estimator can be found in the literature under various names, such as the "multiple model adaptive estimator" [33], the "parallel processing algorithm" [2], the "filter bank method" [7], the "partitioned filter" [11, 19], the "self-tuning estimator" [7], and the "modified Gaussian sum adaptive filter" [8]. These names suggest the structure, features, and capability of this "first-generation" hybrid estimator [20]. This method, however, cannot handle systems with (frequent) mode jumps because its individual model-based filters do not interact with each other.

To overcome the above estimator's inability to handle systems with (frequent) mode jumps, several algorithms were developed, such as the generalized pseudo-Bayesian (GPB) estimators [1, 9, 15], in which the system mode is more realistically (and generally) assumed to be able to jump between members of a mode set, usually modeled as Markovian switching between models. These algorithms, however, are not cost-effective enough for most hybrid estimation problems, regardless whether with a small or large mode set.

The development of the highly cost-effective Interacting Multiple-Model (IMM) estimator [6] was a monumental advance in MM estimation. It is the first estimator that is general and cost-effective enough to be practically applicable to most hybrid (continuous-discrete) estimation problems with a small mode set. Since its development, numerous publications have appeared reporting its successful applications to a variety of important hybrid estimation problems with a small mode set [20].

Most existing MM estimators, including the IMM estimator, have a fixed structure (FS) in the sense that they use a fixed set of models at all times. They have found great success in solving many hybrid estimation problems (i.e., state estimation problems compounded with structural or parametric uncertainty) in many areas over the last three decades, in particular target tracking (see [4, 5, 20] and the long lists of references therein). Thrust by the tremendous success of the IMM estimator, most current research efforts have focused on three fronts: 1) to develop better real-time implementable FS MM estimators; 2) to design better model sets for the particular problems of interest; and 3) to apply MM estimation techniques to solve real-world problems.

It becomes more and more clear from the above efforts that substantial improvement can no longer be expected within the FS and most real-world hybrid estimation problems have a large mode set and cannot be handled satisfactorily by the FS. Ad hoc remedies were proposed for particular applications but few theoretical attempts were made to break away from the FS. The investigation of the moving-bank MM estimators in [34] was an early meaningful effort to go beyond the FS. A serious attempt was initiated in [24] and continued in [22, 25] to lay down a theoretical foundation for MM estimation without the limitation of the FS. Specifically, it formulates the MM estimation in a more general framework of variable structure (VSMM) to overcome the FS estimators' fundamental limitations and to increase the cost-effectiveness. Other theoretical results fundamental to the MM estimation with a variable structure (VS) were also presented in [22, 25], which include the optimal VSMM estimator, a graph-theoretic formulation of MM estimators, and a test criterion for model-set selection. This work shows theoretically the superiority of VS and points out a promising new research direction, in contrast to the existing effort of developing better FS algorithms.

More and more researchers are convinced that VS is probably the *main practical approach* that has the potential to make MM estimators cost-effective enough for real-world problems with a large mode set.

As manifested by the great impact of the monumental success of the IMM estimator on the application of MM estimation techniques, it is fair to say that no matter how promising VSMM estimation may appear, its ultimate success relies on the development of good practical VSMM algorithms that can be readily implemented and are general enough to be applicable to a large class of hybrid estimation problems (i.e., those with both continuous and discrete uncertainties) [20]. If successful, such development will be a new milestone in MM estimation. A primary value of theoretical work on the VS is to facilitate this development.

VSMM estimation is rapidly gaining momentum. Several more or less *ad hoc* VS algorithms/designs have been presented in [10, 16–18, 29–31, 35–37] for the particular applications considered over the past few years. They all fall into either the digraph switching or adaptive grid scheme outlined in Part I of this series [24, 25].

In particular, the first *general* VSMM algorithm has been developed, designed and evaluated in Parts III and IV [29, 30]. It is uncomplicated, applicable to many hybrid estimation problems, and substantially more cost-effective than the IMM estimator. Its successful development relies heavily on the fairly satisfactory theoretical results concerning several important issues in VSMM estimation presented in Part II [21]. They include an efficient and near optimal estimation cycle for VSMM estimation, an efficient and optimal fusion formula, and optimal sequential tests for model-set termination. These results are also essential for the development of the new VSMM algorithm presented here.

A VSMM estimator, called likely-model set (LMS) algorithm, that is generally applicable to most hybrid estimation problems and is fairly easy to implement is presented here. It belongs to one of the three VSMM schemes outlined in [25]—the active digraph scheme. The basic idea of this algorithm is to use the set of all models that are not unlikely at any given time. Several versions are given. The simplest is based on the following idea: Classify all models in effect as either unlikely, significant, or principal by their probabilities (e.g., $P\{m_i\} < t_1, \in [t_1,t_2], > t_2$, respectively). Then model-set adaptation is done by: 1) discard the unlikely ones; 2) keep the significant ones; and 3) activate the models adjacent *"from"* the principal ones. A model is adjacent from another if the former may be switched from the latter. As such, the inclusion of the models adjacent from the principal ones almost surely guarantees superior performance since future mode transition is anticipated; while the elimination of the unlikely modes leads to substantial reduction in computation without performance deterioration.

Simulation results for a target tracking example indicate that the LMS algorithm is substantially more cost-effective than the model-group switching (MGS) algorithm of VS [29, 30] as well as the FS IMM estimator, especially when the total model set is large.

The remaining parts of the paper are organized as follows. Section II formulates the problem and describes briefly the general problem of VSMM estimation. The LMS estimation algorithm is presented in Section III. Section IV covers some practical parts of the LMS algorithm for the examples considered in this work. A simple example of fault detection and identification is simulated in Section

V. Simulation results of an example of tracking a maneuvering target are given in Section VI. These two examples demonstrate the superiority of the LMS algorithm to the FS IMM and the VS MGS algorithms. The last section provides concluding remarks.

II. PROBLEM FORMULATION AND VARIABLE-STRUCTURE MM ESTIMATION

For simplicity of presentation, consider one of the simplest stochastic hybrid systems, known as the (time-varying) "jump linear systems,"

$$x_{k+1} = F(s_{k+1},k)x_k + G(s_{k+1},k)w(s_{k+1},k) \tag{1}$$

$$z_k = H(s_k,k)x_k + v(s_k,k) \tag{2}$$

where x is the (conventional) *base state* vector; z is the noisy measurement vector; s_k is the *modal state* or *system mode* (system mode index) at time k, which denotes the mode in effect during the sampling period ending at t_k; w and v are the mode-dependent process and measurement noise sequences, respectively. The system mode sequence $\langle s_k \rangle$ is assumed to be a first-order homogeneous Markov chain with the transition probabilities

$$P\{s_{k+1} = m_j \mid s_k = m_i\} = \pi_{ij} \qquad \forall \quad m_i, m_j \in \mathbf{S} \tag{3}$$

where $\mathbf{S}$ is the set of all possible system modes.

In the MM approach, it is assumed that the nonlinear system (1)–(2) at any given time can be adequately approximated by one or (a combination of) more models in a set $\mathbf{M}$ consisting of the following M conventional linear models:

$$x_{k+1} = F_k^j x_k + G_k^j w_k^j \qquad \forall \quad m_j \in \mathbf{M} \tag{4}$$

$$z_k = H_k^j x_k + v_k^j \qquad \forall \quad m_j \in \mathbf{M} \tag{5}$$

and a jump between the system modes can be modeled by switching from one model to another, governed by, say, the Markov law (3). Here superscript j denotes quantities pertaining to model m_j. As a result, the MM approach fits well into problems that can be characterized by structural as well as parametric uncertainties and/or changes.

Although other optimality criteria may be used [12], the state estimate and its associated covariance matrix are usually calculated in an MM estimator using the minimum mean square error (MMSE) criterion as follows:

$$\hat{x}_{k|k} = \sum_j \hat{x}_{k|k}^j P\{\mathbf{H}_j^k \mid z^k\} \tag{6}$$

$$P_{k|k} = \sum_j [P_{k|k}^j + (\hat{x}_{k|k} - \hat{x}_{k|k}^j)(\hat{x}_{k|k} - \hat{x}_{k|k}^j)'] P\{\mathbf{H}_j^k \mid z^k\} \tag{7}$$

where $\hat{x}_{k|k}^j$ is the optimal estimate at time k under the hypothesis $\mathbf{H}_j^k \triangleq \{$model sequence j matches the sequence of the true system modes through time $k\}$, and $P_{k|k}^j$ is the associated covariance; $z^k = \langle z_\kappa \rangle_{\kappa \leq k}$ is the measurement sequence through time k with z_0 denoting the initial information. The summations in (6)–(7) of a full-hypothesis-tree (FHT) MM estimator are over all possible hypotheses (i.e., model sequences). Due to the exponential increase in computation and memory of the FHT estimator, one of the following approximate hypothesis management techniques is usually used in practice to limit the growth in the number of hypotheses: 1) elimination of the hypotheses unlikely to be true, which yields a partial-hypothesis-tree (PHT) algorithm, 2) merging of the hypotheses having a common history, which yields a merged-hypothesis-tree (MHT) algorithm, and 3) a combination of hypothesis elimination and merging, which yields a reduced-hypothesis-tree (RHT) algorithm.

Almost all existing MM algorithms have a fixed structure (FSMM) in the sense that the model set $\mathbf{M}$ in (4)–(5) is assumed time invariant, even though the models themselves may be time varying or adaptive. A VSMM estimator is one with a time-varying set $\mathbf{M}$ of models.

The FSMM approach has certain inherent limitations, which stem from its fundamental assumption that the system mode at any time can be represented (with a sufficient accuracy) by one of a fixed set of models that can be determined before measurements are received in real time, and its inability to incorporate certain types of a priori information [37]. The FSMM estimators perform reasonably well for problems that can be handled with a small set of models. When they are applied to solve real-world problems, however, it is often the case that use of only a few models is not good enough. The computational complexity increases considerably as the number of models increases. More importantly, as shown in Part I [22, 25], the use of more models does not necessarily improve the performance; in fact, the performance will deteriorate if too many models are used due to the excessive "competition" from the "unnecessary" (excess) models. Thus one may face a dilemma: more models have to be used to improve the accuracy, but the use of too many models will degrade the performance, let alone the increase in computation. It seems the only way out of this dilemma is the use of a VS as proposed in Part I [24, 25]. Not only can a VSMM estimator take advantage of the real-time system mode information contained in the measurement sequence, but it also is able to incorporate certain a priori information of the system mode that would be difficult or impossible for an FSMM estimator to include. An example of such a priori information is that the system mode is unlikely (but not impossible) to be in a certain subset of the total mode set (mode space) for the particular application under consideration. Another example is the a priori knowledge of the magnitude of jump in system mode.

As shown in Part I [22, 25], a key difference between the optimal VSMM and FSMM estimators is that the former is a probabilistically weighted sum of all estimators based on *admissible mode-set* sequences,[1] rather than of all estimators based on *possible mode* sequences as in the latter. Although the probabilistically weighted sum of all estimators based on the admissible *mode-set* sequences, as required by the optimal VSMM estimator, is computationally infeasible, a practical VSMM estimator may take advantage of its suggested two-level hierarchical structure: multiple *model-set* sequences at the higher level and multiple *model* sequences at the lower level. For many applications, the higher level with multiple model-set sequences should be replaced, due to computational constraints, with a single model-set sequence (most likely along with a proper merging of model-set sequences) obtained by model-set adaptation, which is hopefully one of the best sequences. This is the recursive adaptive model-set (RAMS) approach.

The RAMS approach has two functional components: model-set adaptation and model-sequence conditioned estimation. Part II [21] deals with both components in a general setting. A fairly satisfactory solution to the problem of the model-sequence conditioned estimation was presented. For the model-set adaptation, however, it seems that no solutions may be found that are complete and generally applicable. Nevertheless, the theoretical results presented therein are important in that they provide general principles and guidelines. In fact, a major component of the algorithm proposed here is an application of some relevant theoretical results given there.

The following three practical VSMM schemes were proposed and outlined in [25]: active digraph, digraph switching, and adaptive grid schemes. The algorithm of this paper belongs to the active digraph scheme, which is quite general and powerful.

In the *adaptive grid scheme*, the space of the parameters that characterize the possible system modes is quantized (i.e., represented by a set of models) unevenly and adaptively. A coarse grid is set up initially and then the grid is adjusted recursively according to an adaptation scheme based possibly on the current estimates, model probabilities, model likelihoods, and/or measurement residuals. This approach is particularly advantageous in cases where the set of possible system modes is large. In this scheme, the total model-set need not (and usually cannot) be specified in advance. Designs of such algorithms were reported in [13, 14, 16, 17, 34]. It was shown in [17] that an adaptive grid MM algorithm can provide a substantial improvement in performance at a greatly reduced computational complexity for a maneuvering target tracking problem. While following the same idea of making grid adaptive, different adaptation rules were proposed in different designs for applications characterized by different parameter regions. While valuable, they have a common weakness—they are more or less *ad hoc* and are not generally applicable.

The algorithms presented in [10, 17, 30, 31] belong to the simple and efficient *digraph switching scheme*. In [31], a so-called selected filter IMM design was proposed. Simulation results showed that the proposed design with a substantial reduction in computation yields almost as good performance as that of the FS IMM algorithm for a maneuvering target tracking problem, similar to the one considered here. In [17], a digraph-switching (as well as an adaptive-grid) IMM design for a maneuvering target tracking problem was developed based on coordinated turn models. It was shown that great improvement in performance as well as computation can be achieved by the proposed design relative to the FS IMM algorithm. These designs are, however, *ad hoc* and thus valid only for the problems considered in that no general techniques were proposed for, e.g., when and how a model subset should be switched to another one and how new models should be initialized.

A layered IMM architecture was proposed in [10] for maneuvering target tracking in the presence of glint noise under a fundamental assumption that model sets are independent. It is theoretically equivalent to the IMM algorithm but may have a reduced computational complexity. This algorithm belongs to the soft switching of model subsets. A more general and effective layered IMM algorithm was proposed in [26] without the above independence assumption, which is, however, not really a VS.

The first *general* VSMM algorithm was proposed in Part III [30]. It assumes that the total set of models can be covered by a number of model groups, each representing a cluster of closely related system behavior patterns or structures, and a particular group is running at any given time determined by a two-stage (activation and termination) switching between model groups. How this algorithm can be effectively designed for a particular application was demonstrated in Part IV [29]. Simulation results showed that it is significantly more cost-effective than the IMM estimator, which is the most cost-effective FSMM algorithm developed so far.

[1] A mode-set sequence is admissible if it is one that may contain the true mode sequence as its element.

Compared with the above algorithms/designs, the superiority and difference of the proposed LMS algorithm can be summarized as follows.

1) None of the above algorithms/designs belong to the active digraph scheme while the proposed LMS algorithm does.

2) The LMS algorithm can be applied to a larger class of hybrid estimation problems than the MGS algorithm, which is much more general than all the other VSMM algorithms/designs.

3) The design of the LMS algorithm is much simpler than that of the MGS algorithm: In addition to the design elements of an FSMM estimator, only two thresholds need to be tuned in the LMS algorithm while the design of the model groups and the activation logics, as well as termination thresholds are necessary in the MGS algorithm. This difference may be crucial in practice.

4) Our simulation results indicate that the LMS algorithm is more cost-effective than the MGS algorithm.

III. LIKELY-MODEL SET MM ESTIMATOR

Note that an FS full-hypothesis-tree MM estimator is not necessarily optimal because the model set used in general does not match the set of all possible system modes: $\mathbf{M} \neq \mathbf{S}$.

Adaptation of model set has two opposite tasks: reduction and expansion of the model set.

The basic idea of the proposed LMS algorithm[2] is to use at any given time a subset of the models in the total set that are not unlikely to be in effect at the given time.

The key in the implementation of the above idea is the concept of the state dependency of the mode set, as introduced in Part I [22, 25]. In plain terms, it implies that given the current system mode, the set of possible system modes at the next time is a subset of the total mode set, which is determined by the Markovian mode transition law (i.e., the graph-theoretic adjacence of the mode sets). Mathematically, the state-dependent system mode set at time $k+1$ with respect to (wrt) the previous hybrid state $\xi_k \stackrel{\Delta}{=} (x_k, m_k)$ is defined by

$$S^{\xi_*}_{k+1} = \{m : P\{s_{k+1} = m \mid \xi_*\} = \phi[k, \xi_*, m] > 0\} \tag{8}$$

where $P\{s_{k+1} = m \mid x_*\}$ is a more general version of (3). A closely related but slightly simpler concept is the mode dependency of mode set. The mode-dependent system mode set wrt rnode m_* is defined by

$$S^{m_*}_{k+1} = \{m : P\{s_{k+1} = m \mid s_k = m_*, x_k\} > 0 \text{ for some } x_k\}. \tag{9}$$

Note that $S^{\xi_*}_{k+1}$ is a subset of $S^{m_*}_{k+1}$.

To derive the LMS algorithm, assurne that the initial system mode s_0 is known a priori to be in $S_0 \subset \mathbf{S}$, where S_0 is known (set it equal to $\mathbf{S}$ if S_0 is not known). Note that if $s_0 = m_*$, then the system mode set at the next time ($k = 1$) would be $S^{m_*}_1$. Since s_0 could be any element of S_0, the set of possible system modes at time $k = 1$ is the union of all the mode-dependent mode sets wrt all the elements of S_0, given by

$$S_1 = \bigcup_{m \in S_0} S^m_1. \tag{10}$$

This process can be repeated with the following recursion for k:

$$S_k = \bigcup_{m \in S_{k-1}} S^m_k. \tag{11}$$

Consequently, if an MM estimator is the optimal one given the model set, it would be overall optimal using the model-set sequence $M^k = S^k$, where $S^k = \langle S_\kappa \rangle_{\kappa \leq k}$ is determined by (11). In reality, however, the model set so determined would expand rapidly in size and become equal to the total set $\mathbf{S}$ soon, which is usually unknown or too large. This may be computationally too expensive. This is because there is no model-set reduction mechanism in the above model-set sequence determination.

In view of this and the fact that the total model set $\mathbf{M}$ used in practice usually does not match exactly the set $\mathbf{S}$ of all possible system modes, a suboptimal algorithm is to use at each time k a subset of the total *model* set[3] $\mathbf{M}$ that has a good chance of capturing the system mode in effect at k. Clearly, this implies that models unlikely to match the system mode in effect at the time considered are eliminated from the "working" subset. This is the underlying idea of the LMS algorithm.

The question now is how to measure the chance that a subset M_* of the total set $\mathbf{M}$ captures the system mode in effect at the time considered. Two methods are available in principle, based on the model-set (marginal) likelihood $p(z_k \mid s_k \in M_*, z^{k-1})$ and the model-set probability $P\{s_k \in M_* \mid s_k \in \mathbf{M}, z^k\}$, respectively. It was shown in Part II [21] that the model-set probability and likelihood are given by

$$P\{s_k \in M_* \mid s_k \in \mathbf{M}, z^k\} = \sum_{m \in M_*} P\{s_k = m \mid s_k \in \mathbf{M}, z^k\} \tag{12}$$

[2] Unfortunately, our proposed algorithm and the well-known Least Mean Square algorithm in adaptive signal processing have the same abbreviation. In the case where a distinction between them is needed or desired, the former is abbreviated as LMS-MM algorithm.

[3] Not total *mode* set $\mathbf{S}$, which could be unknown or too large.

$$p(z_k \mid s_k \in M_*, z^{k-1}) = \sum_{m \in M_*} p(z_k \mid s_k = m, z^{k-1}) \times \frac{P\{s_k = m \mid s_k \in \mathbf{M}, z^{k-1}\}}{P\{s_k \in M_* \mid s_k \in \mathbf{M}, z^{k-1}\}} \quad (13)$$

where $P\{s_k = m \mid s_k \in \mathbf{M}, z^k\}$ and $p(z_k \mid s_k = m, z^{k-1})$ are model probability and marginal likelihood, respectively. Note that the model-set *joint* likelihood $p(z^k \mid s_k \in M_*)$ is not appropriate here because the model set is variant. Of course, more accurate measures are obtained if $p(z_k \mid s_\kappa \in M_\kappa,\ \kappa \leq k)$ is used or if the total model-set $\mathbf{M}$ in the above is replaced with the total mode set $\mathbf{S}$, which is however infeasible.

Note that a model m_* that cannot be jumped from any model in the set M_{k-1} used for the previous time should not be included[4] in the model set M_k; that is, if $m_* \notin \bigcup_{m \in M_{k-1}} S_k^m$, then $m_* \notin M_k$. In other words, every model in M_k can be jumped from some model(s) in M_{k-1} and thus its a priori probability before time k is available. As a result, containing information from previous time, the set probability is normally a better measure than the set likelihood. However, as shown in Part II [21], if the LMS estimator has to be initialized without prior distribution of model probabilities, then the set likelihood is equivalent to the set probability assuming equal prior model probabilities.

LMS1 Algorithm

S1. Model-set expansion: Obtain the union of the system mode sets $M_k = \bigcup_{m \in M'_{k-1}} S_k^m$, where S_k^m is the mode-dependent system mode set wrt m, defined by (9), and M'_{k-1} is defined in Step 3 below.

S2. Model-set sequence conditioned estimation:

a) Initialization: Obtain predicted probability of each model in M_k and initialize all those filters based on the newly activated models.

b) Mode-matched estimation: For each model in M_k, obtain estimates under the assumption that this model matches the system mode in effect exactly, as in an FSMM estimator using M_k.

c) Model probability evaluation: Evaluate the probability of each mode in M_k, as in an FSMM estimator.

d) Estimate fusion: Obtain the overall estimate and its associated covariance as in an FSMM estimator.

S3. Model-set reduction (unlikely model elimination): Form M'_k as the subset of M_k by deleting all its unlikely models.

The following two methods may be used for the elimination of unlikely models in the set M_k.

Delete all the models in M_k except the K models of the largest probabilities, where K is a constant, determined on the basis of computational considerations.

Rank all the models in M_k by their model probabilities and delete those whose ratio of model probability to the largest one is below a certain threshold. This is more general and requires less tuning of the threshold than thresholding the model probabilities directly (i.e., rather than the ratio). This can be justified by the sequential ranking test of Part II.

Note that the first method is equivalent to the second one with a time-varying threshold such that the number of remaining models is K. Thus, the first method has fixed computation but variant "performance" while the second has fixed "performance" but variant computation.

The following combinations of these two methods may be more reasonable for certain applications.

"AND" combination: A model is deleted only if its probability is below the threshold and not among the largest K model probabilities. This leads to at least K models that are not deleted.

"OR" combination: A model is deleted if its probability is either below the threshold or not among the largest K model probabilities. This leads to at most K models that are not deleted.

The AND combination seems more reasonable in most cases because it guarantees "performance" and relaxes computation while OR combination guarantees computation and relaxes "performance."

The model probability evaluation step above can be done based on the following

$$P\{m_k \mid m_{k-1}, M_k, z^k\} = \frac{1}{c} p[z_k \mid m_k, m_{k-1}, z^{k-1}] \times P\{m_k \mid m_{k-1}, M_k, z^{k-1}\} \quad \forall \ m_k \in M_k \qquad (14)$$

where $c = p[z_k \mid m_{k-1}, M_k, z^{k-1}]$ is a normalization factor; $p[z_k \mid m_k, m_{k-1}, z^{k-1}]$ is the likelihood function; and $P\{m_k \mid m_{k-1}, M_k, z^{k-1}\}$ is the predicted mode probability, which can be obtained from $P\{m_{k-1} \mid z^{k-1}\}$ and M_k. Note that M_k is used for notational simplicity to denote the model set, along with the mode transitions governed by a (state-dependent) Markov chain.

To further reduce computation, elimination of the unlikely models can be done based on the predicted model probabilities $P\{m_k \mid m_{k-1}, M_k, z^{k-1}\}$ rather than the posterior model probabilities $P\{m_k | m_{k-1}, M_k, z^k\}$. This leads to the following algorithm.

LMS2 Algorithm

S1. Model-set expansion: Obtain M_k^* exactly the same as M_k of the LMS1 algorithm.

S2. Model-set reduction (unlikely model elimination): Form M_k as the subset of M_k^* by

[4] Although it is still possible (but with a small probability) that it matches the system mode in effect at k.

deleting all its unlikely models on the basis of the predicted model probabilities.

S3. Model-set sequence conditioned estimation: Same as the LMS1 algorithm.

Note that the use of the predicted instead of posterior model probabilities reduces substantially the size of the "working" set M_k: M_k of the LMS2 algorithm does not include unlikely models whereas M_k of the LMS1 algorithm does.

The LMS2 algorithm can be further simplified based on the following idea. It is usually desirable to classify all models into three categories: unlikely, significant and principal. Consequently, a reasonable set of rules for model-set adaptation is: 1) discard the unlikely ones; 2) keep the significant ones; and 3) activate the models adjacent *from* the principal ones. A model m_j is adjacent from m_i if the transition probability from m_i to m_j is not zero. This leads to the LMS3 algorithm in which the model-set adaptation has the following recursive relation:

$$M_k = (M_{k-1} - U_{k-1}) \cup \left(\bigcup_{m_i \in M_{k-1}^p} A_{m_i} \right) \tag{15}$$

where U_{k-1} is the set of unlikely models in M_{k-1}; M_{k-1}^p is the set of principal models in M_{k-1}; and A_{m_i} is the set of models that are adjacent *from* model m_i, defined by

$$A_{m_i} = \{m_j \in \mathbf{M} : P\{s_k = m_j \mid s_{k-1} = m_i\} \neq 0\}. \tag{16}$$

Note that

$$\begin{aligned} M_k &= (M_{k-1} - U_{k-1}) \cup \left(\bigcup_{m_i \in M_{k-1}^p} A_{m_i} \right) \\ &\neq M_{k-1} \cup \left(\bigcup_{m_i \in M_{k-1}^p} A_{m_i} \right) - U_{k-1}. \end{aligned} \tag{17}$$

That is, the unlikely models are deleted first, then the models adjacent from some principal models are activated and thus those unlikely models that are adjacent from a principal model will be retained.

LMS3 Algorithm

S1. Model classification: Identify each mode in M_{k-1} to be unlikely (if its probability is below t_1), principal (if its probability exceeds t_2) or significant (if its probability is in between t_1 and t_2).

S2. Model-set adaptation: Obtain M_k according to (15). In plain words, this consists of the following two steps:

a) Unlikely model elimination: Delete all the unlikely models of M_{k-1}.

b) Possible model activation: Activate all the models adjacent from some principal model of M_{k-1}.

S3. Model-set sequence conditioned estimation: Same as the LMS1 algorithm.

Note the differences between the LMS3 and LMS2 algorithms. a) The models that are adjacent from a significant model but not from a principal model may be activated in the LMS2 but not in the LMS3 algorithm. b) The models that are adjacent from a principal model are always activated in the LMS3 algorithm but may not be activated in the LMS2 algorithm if deemed unlikely.

Table I gives one cycle of the LMS3 algorithm with the AND logic for the elimination of the unlikely models. In fact, the model classification in Step 2 can be done based on the predicted model probabilities. Step 3 is based on the optimal fusion of [21, Theorem 4]. VSIMM cycle is given in [21].

It should be emphasized that the LMS estimator is general and needs only to tune the two thresholds t_1 and t_2, which is much simpler than the design of model groups and activation and termination logics as in the MGS algorithm. It seems that the only potential drawback of the LMS estimator is its weakness in handling jumps between two widely separated modes that are connected only through several intermediate modes in the total mode set. Such jumps occur rarely or the topology of the total model set was designed improperly. A possible cure or alleviation of this weakness is by repeatedly applying the three adaptation rules in each time step until nothing changes. For example, if a newly activated model turns out to be a principal one *at the time when it is activated* according to the *posterior* probabilities, then activate the models adjacent from it *immediately*, and this process can be repeated within one time step (note that every model probability may change once the model set varies).

Similar to the MGS estimator, since there is no guarantee that the model-set adaptation is done correctly and timely, such adaptation may introduce errors in the state estimate and the associated covariance of the filters based on the newly activated models. Three remedies may be used. A good and systematic technique is to go back several steps in time to initialize the newly activated models and their filters. If a model is activated at k, then n-step back means that it has zero probability at $k-n-1$ but in general non-zero probability at $k-n$. Thus, we may run the generic VSIMM recursion of Part II [21] several times, VSIMM$[M_{k-n}, M_{k-n-1}]$, VSIMM$[M_{k-n+1}, M_{k-n}], \ldots,$ VSIMM$[M_k, M_{k-1}]$, to obtain the current estimates of the model. Of course, old overall estimate (and everything that has already been sent out) may not be overridden unless the

TABLE I
One Cycle of LMS3 Algorithm with AND Logic

S1. Increase the time counter k by 1. Run the VSIMM$[M_k, M_{k-1}]$ cycle.

S2. Classify all the models m_i's in M_k to be principal (i.e., $\mu_k^i > t_2$), unlikely (i.e., $\mu_k^i < t_1$), or significant (i.e., $t_1 \leq \mu_k^i \leq t_2$). Let the set of unlikely models be M_u. If there is neither unlikely nor principal model, then output $\hat{x}_{k|k}$, $P_{k|k}$, and $\{\mu_k^i\}_{m_i \in M_k}$, let $M_{k+1} = M_k$ and go to Step 1.

S3. If there is no principal model, then let $M_a = \emptyset$ and go to Step 4. Otherwise, identify the set M_a of all the models adjacent to any principal model. Find the set of new models $M_n = M_a \cap \overline{M}_k$ (where $\overline{M}_k$ is the complement of M_k) and the union set $M_k := M_n \cup M_k$. Then

- Run the VSIMM$[M_n, M_{k-1}]$ cycle, where M_n is the set of new and only new models.
- Fusion: Calculate the estimates, error covariances, and mode probabilities for the union set M_k:

$$\mu_k^i = \frac{L_k^i \hat{\mu}_{k|k-1}^i}{\sum_{m_i \in M_k} L_k^i \hat{\mu}_{k|k-1}^i}, \qquad \forall \quad m_i \in M_k$$

$$\hat{x}_{k|k} = \sum_{m_i \in M_k} \hat{x}_{k|k}^i \mu_k^i$$

$$P_{k|k} = \sum_{m_i \in M_k} [P_{k|k}^i + (\hat{x}_{k|k}^i - \hat{x}_{k|k})(\hat{x}_{k|k}^i - \hat{x}_{k|k})'] \mu_k^i$$

where the estimates $\{\hat{x}_{k|k}^i\}$, error covariances $\{P_{k|k}^i\}$, likelihoods $\{L_k^i\}$, and predicted probabilities $\{\hat{\mu}_{k|k-1}^i\}$ were obtained in the above VSIMM$[M_k, M_{k-1}]$ and VSIMM$[M_n, M_{k-1}]$ cycles.

S4. Output $\hat{x}_{k|k}$, $P_{k|k}$, and $\{\mu_k^i\}_{m_i \in M_k}$.

S5. If there is no unlikely model, go to Step 1; otherwise, identify the discardable model set $M_d = M_u \cap \overline{M}_a$; that is, the set of unlikely models that are not adjacent from any principal model.

S6. Eliminate the models in M_d from M_k that have smallest probabilities such that M_k has at least K models; that is, let the likely-model set be $M_l = M_k - M_m$, where M_m is the set of models in M_d with smallest probabilities such that M_l has at least K models.

S7. Let $M_{k+1} = M_l$. Go to Step 1.

problem at hand is smoothing, instead of filtering. One-step back is recommended and was used in our examples.

A theoretically more appealing technique is to replace the VSIMM$[M_k, M_{k-1}]$ of zero memory depth with a generic VSMM$[M_k, M_{k-1}, \ldots, M_{k-n-1}]$ cycle of n memory depth. This, however, requires the implementation of such a VSMM cycle.

A less systematic technique is to increase all the error covariances $P_{k-1|k-1}^n$ of the filters based on a newly activated model to some degree (e.g., by using a fudge, forgetting, or discount factor) to account for the extra uncertainties in the estimates of these filters. For example, a time-varying forgetting (discount) factor λ_k may be used for all newly activated models as follows:

$$P_{k-1|k-1}^j := P_{k-1|k-1}^j / \lambda_{k-1}, \qquad m_i \in (M_k \cap \overline{M}_{k-1}) \tag{18}$$

where $\overline{M}_{k-1}$ is the complement of M_{k-1} and λ_k may be given by

$$\lambda_k = 1 - \lambda^o + \lambda^o \lambda_{k-1} \tag{19}$$

and $0 < \lambda_0$, $\lambda^o < 1$ are design parameters that control the initial and variation rate of the forgetting factor. Note that this technique does not alter the estimates directly.

IV. DESIGN OF LMS ALGORITHM AND PERFORMANCE MEASURES

A. Design of LMS Algorithm

In the sequel, MGS stands for the MGS algorithm developed in Part III [30] and designed in Part IV [29] using IMM algorithm for the model-set conditioned estimation; IMM stands for the FSIMM estimator, with a lower bound 10^{-300} on the mode transition probabilities [28]; GPB1 and GPB2 are the abbreviations of the first- and second-order generalized pseudo-Bayesian estimators, respectively; SMM denotes the nonswitching (static) MM estimator, with a lower bound 10^{-3} on the mode probabilities [28]; LMS stands for the LMS3 algorithm of Table I using IMM algorithm for the model-set conditioned estimation; the corresponding algorithm with a time-varying forgetting factor with $\lambda^o = 0.70$ and $\lambda_0 = 0.30$ is denoted as LMS(λ). The forgetting factor is activated whenev er the estimated acceleration magnitude exceeds a threshold; that is,

$$\hat{a}_k = \sqrt{\left(\frac{\hat{v}_k^x - \hat{v}_{k-1}^x}{T}\right)^2 + \left(\frac{\hat{v}_k^y - \hat{v}_{k-1}^y}{T}\right)^2} > \sqrt{2r}$$

$$\Rightarrow \text{activate forgetting factor} \tag{20}$$

where $[\hat{v}_k^x, \hat{v}_k^y]' = [\hat{v}_{k|k}^x, \hat{v}_{k|k}^y]'$ is the estimated velocity vector; T is the sampling period and r is the variance of measurement noise along each axis.

The two thresholds for model classification were set at $t_1 = 10^{-4}$ and $t_2 = 0.3$ for all examples considered. An AND combination of the two methods for eliminating unlikely models was used, where K was chosen to be 5 for topology A and 9 for topology B.

Using the VSIMM cycle of [21] there is no need to design the transition probability matrix for each model set in the LMS and MGS algorithm given the transition probability matrix for the total model set. The VSIMM cycle will automatically use the right submatrix of this transition probability matrix and scale appropriately to preserve the relative ratios of the transitions within the set. All these are done naturally and systematically. See Part II [21] for details.

B. Performance Measures

The three most essential components of the evaluation of an MM estimator are its state estimation quality (e.g., rms position and velocity errors), mode identification capability (e.g., mode probabilities) and computational complexity (e.g., flops and CPU time). General considerations and specific criteria for such an evaluation were given in [27]. Other components may include robustness, parallelism, implementability, etc.

In this work, some additional measures of the mode identification capability were used for the evaluation of the proposed LMS algorithm. They are the same as in Part IV [29] for the MGS algorithm. For the reader's convenience, they are described briefly here. For details, the reader is referred to [27, 29]. These measures are as follows.

1) the percentages of correct mode identification (CID), incorrect mode identification (IID), and no mode identification (NID):

a) a CID is obtained if the model closest to the system mode in effect at the given time has the highest probability that exceeds a threshold (say 0.5);

b) an IID is obtained if the model with the highest probability that exceeds the threshold is not closest to the system mode in effect at the given time;

c) it is indecisive (NID) if no model has a probability above the threshold.

2) the average distances (over N Monte Carlo runs) between the system mode s_k in effect at time k and the models m_i used in the MM algorithm at k, given by

$$\text{average modal distance at } k = \frac{1}{N}\sum_{n=1}^{N}\sum_{m_i \in M_k} \|s_k - m_i\| P\{s_k = m_i \mid s_k \in M_k, z^k\} \tag{21}$$

where $\|s_k - m_i\|$ is the Euclidean distance.

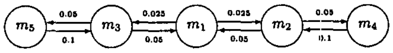


Fig. 1. Topology of total model set.

3) The rms error of mode estimation and the average rms error, defined by

$$\text{rms mode error at } k = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\|s_k - \hat{s}_{k|k}\|^2} \tag{22}$$

$$\text{average rms error} = \frac{1}{K}\sum_{k=1}^{K}(\text{rms error at } k) \tag{23}$$

where the mode estimate is defined by

$$\hat{s}_{k|k} = \sum_{m_i \in M_k} m_i P\{s_k = m_i \mid s_k \in M_k, z^k\}. \tag{24}$$

All simulation results presented in this paper are over 500 Monte Carlo runs.

V. EXAMPLE OF FAULT DETECTION AND IDENTIFICATION

A simple example of fault detection and identification was simulated as presented in [30].

Consider the following simple scalar system with 2-dimensional measurements

$$x_{k+1} = 1.1x_k + w_k \tag{25}$$

$$z_k = \begin{bmatrix} 0.5 \\ 1.2 \end{bmatrix} x_k + v_k \tag{26}$$

with $w_k \sim \mathcal{N}(0, 0.1^2)$, $v_k \sim \mathcal{N}(0, I)$. A total failure or partial fault may occur to sensor 1, that is, 0.5 may become 0 (total failure) or any number in between 0 and 0.5 (partial fault). Suppose that sensor 2 may also suffer from a total or partial fault.

Assume that the following 5 models constitute the total model set for the MM approach:

$$m_1 : \begin{bmatrix} 0.5 \\ 1.2 \end{bmatrix} \quad m_2 : \begin{bmatrix} 0.25 \\ 1.2 \end{bmatrix} \quad m_3 : \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix}$$
$$m_4 : \begin{bmatrix} 0 \\ 1.2 \end{bmatrix} \quad m_5 : \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} \tag{27}$$

The digraph (topology of the model set with model transition probabilities) assumed is shown in Fig. 1, where the self-transition probabilities can be obtained from the unity requirement for the sum of all transition probabilities from a model.

The following simple fault scenario was considered. The system has no fault from $k = 0$ to $k = 160$, except that a sensor 2 total failure occurs at $k = 100$ and lasts until $k = 120$.

The fault detection and identification results over the time period considered depend to a large degree on the initial state of the system because

TABLE II
Comparison of LMS, MGS, IMM, GPB2, GPB1, SMM Algorithms

	FLOP Ratio	CID (%)	IID (%)	NID (%)	Modal Distance	Mode Error	RMS Error	Max Error
$x_0 = 100$:								
LMS	0.546	100	0	0	0	0	0.418	0.869
MGS	0.546	100	0	0	0	0	0.418	0.869
IMM	1	100	0	0	0	0	0.418	0.869
GPB2	3.792	100	0	0	0	0	0.418	0.869
GPB1	0.829	100	0	0	0	0	0.418	0.869
SMM	0.795	99.4	0.625	0	0.0065	0.0060	272373	5734169
$x_0 = 10$:								
LMS	0.546	99.9	0.08	0	0.0004	0.0011	0.4214	1.236
MGS	0.546	99.9	0.08	0	0.0004	0.0011	0.4215	1.236
IMM	1	99.9	0.08	0	0.0004	0.0011	0.4214	1.236
GPB2	3.792	99.9	0.08	0	0.0004	0.0011	0.4213	1.236
GPB1	0.829	99.9	0.08	0	0.0004	0.0011	0.4221	1.236
SMM	0.795	99.3	0.70	0	0.0069	0.0069	27260	573897
$x_0 = 1$:								
LMS	0.572	94.9	1.78	3.34	0.0211	0.0239	0.443	1.13
MGS	0.573	94.8	2.24	2.96	0.0228	0.0366	596.5	8448
IMM	1	94.5	1.66	3.79	0.0225	0.0251	0.445	1.18
GPB2	3.792	94.6	1.64	3.74	0.0223	0.0249	0.442	1.18
GPB1	0.829	93.5	2.91	3.6	0.0285	0.0341	0.464	1.07
SMM	0.795	92.8	3.99	3.21	0.0288	0.0301	2801	58967
$x_0 = 0.1$:								
LMS	0.608	84.8	2.10	13.1	0.0668	0.0813	0.4674	1.069
MGS	0.573	89.8	1.98	8.24	0.0518	0.0619	0.4570	1.069
IMM	1	83.1	1.99	14.9	0.0724	0.0829	0.4697	1.116
GPB2	3.792	83.2	1.92	14.8	0.0721	0.0824	0.4669	1.116
GPB1	0.829	80.6	6.75	12.7	0.1070	0.1318	0.5247	0.921
SMM	0.795	80.2	7.97	11.8	0.0738	0.0819	640.41	1348

the noise levels are fixed but the magnitude of the state is dependent of its initial value. Table II gives comparison of the LMS, MGS, IMM, GPB2, GPB1, and SMM algorithms for the fault detection and identification results for the true initial state $x_0 = 0.1$, 1, 10, and 100, respectively. In the table, the MGS algorithm was designed as in [30]. All the switching/interacting algorithms used the same transition probabilities. No decision (detection) delay was observed for any algorithm for this example.

Clearly the LMS algorithm is most cost-effective. Note also that the case with a smaller initial x_0 is more difficult for fault detection and identification since the noise levels are relatively higher. The SMM algorithm has huge base state estimation errors. Note that when $x_0 = 0.1$, the measurements are dominated by the noise.

Fig. 2 shows for $x_0 = 1$ the probabilities of the model that match the true mode at each given time, as calculated in the LMS, MGS, IMM, GPB2, GPB1, and SMM algorithms. Note that after the initial transients, only the SMM algorithm suffers from a significant error in the mode probability around the time when the fault is removed.

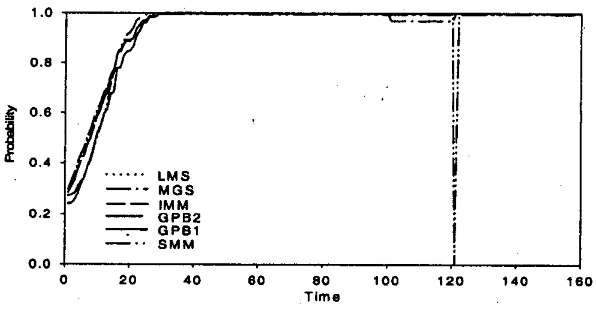


Fig. 2. Average probability of model matching true mode.

VI. EXAMPLE OF MANEUVERING TARGET TRACKING

Following [29], consider the following target tracking example, adopted from [3]. To track a maneuvering target in a planar motion that may have a piecewise constant acceleration with a maximum value of $4g$ (40 m/s^2) in any direction, the performance of several MM estimators was compared in [3, 29, 31, 35] based on the following set of 13 time-invariant models, characterized by the expected acceleration vector **a**,

$$
\begin{array}{ll}
m_1 : \mathbf{a} = [0,0]' & m_2 : \mathbf{a} = [20,0]' \\
m_3 : \mathbf{a} = [0,20]' & m_4 : \mathbf{a} = [-20,0]' \\
m_5 : \mathbf{a} = [0,-20]' & m_6 : \mathbf{a} = [20,20]' \\
m_7 : \mathbf{a} = [-20,20]' & m_8 : \mathbf{a} = [-20,-20]' \\
m_9 : \mathbf{a} = [20,-20]' & m_{10} : \mathbf{a} = [40,0]' \\
m_{11} : \mathbf{a} = [0,40]' & m_{12} : \mathbf{a} = [-40,0]' \\
\multicolumn{2}{c}{m_{13} : \mathbf{a} = [0,-40]'.}
\end{array}
\tag{28}
$$

These models were obtained by a quantization of the possible system mode space in acceleration.

Two graph-theoretic (digraph) representation (see [25]) of these 13 models are given in Fig. 3, where an arrow from one model to another indicates a legitimate model switch and self-loops are omitted (i.e., each model may stay in itself for some time). Fig. 3 was plotted in the model space characterized by the expected acceleration with the following nice features.

1) Each model can be viewed as a point in the model space.
2) Model m_1 corresponds to the origin of the model space.
3) Models m_{12}, m_4, m_1, m_2, and m_{10} form the horizontal axis, while models m_{13},m_5,m_1,m_3, and m_{11} form the vertical axis.
4) The graphical distance between any two models in Fig. 3 is exactly proportional to the actual distance between the two models in their acceleration space.

Note that in topology A, a model is only allowed to switch to its nearest neighbors as well as itself, while in topology B, switches to its second nearest neighbors (e.g., m_2 and m_3, and m_6 and m_{10}) are also allowed.

Experience indicates that the use of (second-order) nearly constant velocity (CV) models [4] with specified expected accelerations led to significantly better results than (third-order) nearly constant acceleration (CA) models [4]. Also, poor results would be obtained if the accelerations of a model were taken as part of the state. Therefore, only CV models with the specified accelerations are considered here.

A. Test Scenarios

It should be emphasized that the evaluation and thus comparison of MM algorithms depend to a large degree on the scenarios used. Both deterministic and random scenarios were designed for this example.

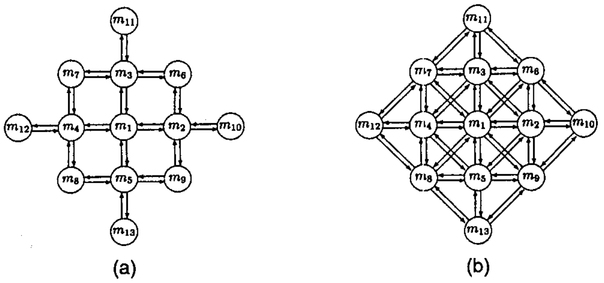


Fig. 3. Two topologies of model-set (28). (a) Topology A. (b) Topology B.

For the random scenario, it is assumed that the acceleration vector $\mathbf{a}(t) = a(t)\angle\theta(t)$ is a semi-Markov process; specifically, it is a 2-dimensional process that would be Markov were the sojourn time τ for each of its states not random. In simple terms, it implies that the acceleration process undergoes sudden jumps from a state with a magnitude a and phase θ to another one after staying in it for a random period of time. Specifically, the following models of **a**, as proposed in [29], was used.

1) The sojourn time τ_k of the state $\mathbf{a}_k$ conditioned on $\mathbf{a}_k$ has a truncated ($\tau_k > 0$) Gaussian density with mean $\bar{\tau}$ and variance σ_τ^2.
2) The acceleration magnitude a_{k+1} has probability masses of P_0 and P_M to be zero and maximum, respectively, and is uniform over the values in between, where P_0 and P_M are in general functions of a_k.
3) The angle θ_{k+1} of acceleration is uniform over 2π if $a_k = 0$ and is Gaussian with mean θ_k and variance σ_θ^2 if $a_k \neq 0$.

The following parameters were used in our design:

$$
\bar{\tau} = \bar{\tau}_M + \frac{a_{\max} - a}{a_{\max}}(\bar{\tau}_0 - \bar{\tau}_M), \qquad
\sigma_\tau = \tfrac{1}{12}\bar{\tau}_a, \qquad \bar{\tau}_M = 10 \quad \bar{\tau}_0 = 30 \tag{29}
$$

$$
P_M = 0.1, \qquad a_{\max} = 37, \qquad \sigma_\theta = \pi/12, \qquad
P_0 = \begin{cases} 0.6 & a_k \neq a_{\max} \\ 0.8 & a_k = a_{\max} \end{cases}. \tag{30}
$$

The random sojourn time τ was rounded to its nearest integer and the initial acceleration a_1 was set to zero.

The use of such a random test scenario reduces the dependence of the performance of an MM algorithm on various artifacts of a scenario. With such a random scenario, it is difficult to design an MM algorithm with subtle tricks that are effective only for certain scenarios. For such a random scenario, however, the peak and steady-state estimation errors are no longer separable when averaged over Monte Carlo runs. For

TABLE III
Deterministic Scenarios (Sequence of Accelerations)

Time Period	1–30	31–45	46–55	56–80	81–98	99–119	120–139	140–150	151–160
Scenario 1: $\mathbf{a}_k$	$[0,0]'$	$[18,22]'$	$[2,37]'$	$[0,0]'$	$[25,2]'$	$[-2,19]'$	$[0,-1]'$	$[38,-1]'$	$[0,0]'$
Scenario 2: $\mathbf{a}_k$	$[-20,0]'$	$[22,22]'$	$[-22,22]'$	$[0,0]'$	$[30,2]'$	$[-2,39]'$	$[0,-20]'$	$[2,40]'$	$[0,0]'$

Note: Second scenario has several large jumps in system mode.

this reason, several deterministic scenarios were also used so that the peak and steady-state errors can be seen well. Due to space limitation, however, only some results for the above random scenario and the following two deterministic scenarios are reported here: The system mode sequence, characterized by the sequence of accelerations $\mathbf{a}_k$, is specified in Table III.

B. Design of Adjacency Index Matrices

A unique feature of the proposed LMS3 algorithm is that it relies quite explicitly on the adjacency relations among the models in the total model set (or more precisely, in the total digraph). A model is said to be adjacent from another if the former may be switched from the latter. Based on the adjacency of the total model set designed for a particular problem, such as topology A and topology B in this paper, it is easy to define an adjacency index matrix to describe the adjacency relations. The following adjacent index matrices Ω were used in the LMS algorithm, for topology A

$$\Omega = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 & 3 & 4 & 5 & 2 & 3 & 4 & 5 \\ 2 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ 3 & 9 & 6 & 7 & 8 & 3 & 4 & 5 & 2 & 10 & 11 & 12 & 13 \\ 4 & 10 & 11 & 12 & 13 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ 5 & 6 & 7 & 8 & 9 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \end{bmatrix} \tag{31}$$

and for topology B,

$$\Omega = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 3 & 4 & 5 \\ 2 & 2 & 3 & 4 & 5 & 2 & 3 & 4 & 5 & 9 & 6 & 7 & 8 \\ 3 & 5 & 2 & 3 & 4 & 10 & 11 & 12 & 13 & 10 & 11 & 12 & 13 \\ 4 & 9 & 6 & 7 & 8 & 6 & 7 & 8 & 9 & 6 & 7 & 8 & 9 \\ 5 & 10 & 11 & 12 & 13 & 11 & 12 & 13 & 10 & 10 & 11 & 12 & 13 \\ 6 & 6 & 7 & 8 & 9 & 3 & 4 & 5 & 2 & 10 & 11 & 12 & 13 \\ 7 & 3 & 4 & 5 & 2 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ 8 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ 9 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \end{bmatrix} \tag{32}$$

In these matrices, the ith column lists all models that are adjacent from model m_i, including m_i itself. The repetitions of the same numbers in a given column is made such that a rectangular matrix, rather than a ragged two-dimensional array, is obtained. Note that these adjacency index matrices are different from the adjacency matrix commonly defined in graph theory.

The adjacency index matrix for the above simple example of fault detection and identification is similarly obtained.

Note that the adjacency index matrix follows immediately from the underlying digraph. Although the matrix is not unique, different matrices corresponding to the same underlying digraph are equivalent in that they lead to identical results. Design of underlying digraph (and the adjacency index matrix) is a basic task of applying any interaction/switching MM estimator, as described in Part I [25].

C. Other Design Elements

The following transition probability matrices were used for all the IMM algorithms considered: For topology A,

$$\Pi = \begin{bmatrix}
116/120 & 1/120 & 1/120 & 1/120 & 1/120 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.02 & 0.95 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0.01 & 0.01 & 0 & 0 & 0 \\
0.02 & 0 & 0.95 & 0 & 0 & 0.01 & 0.01 & 0 & 0 & 0 & 0.01 & 0 & 0 \\
0.02 & 0 & 0 & 0.95 & 0 & 0 & 0.01 & 0.01 & 0 & 0 & 0 & 0.01 & 0 \\
0.02 & 0 & 0 & 0 & 0.95 & 0 & 0 & 0.01 & 0.01 & 0 & 0 & 0 & 0.01 \\
0 & 1/30 & 1/30 & 0 & 0 & 28/30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1/30 & 1/30 & 0 & 0 & 28/30 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1/30 & 1/30 & 0 & 0 & 28/30 & 0 & 0 & 0 & 0 & 0 \\
0 & 1/30 & 0 & 0 & 1/30 & 0 & 0 & 0 & 28/30 & 0 & 0 & 0 & 0 \\
0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9 & 0 & 0 & 0 \\
0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9 & 0 & 0 \\
0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9 & 0 \\
0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9
\end{bmatrix} \tag{33}$$

and for topology B,

$$\Pi = \begin{bmatrix}
348/360 & 2/360 & 2/360 & 2/360 & 2/360 & 1/360 & 1/360 & 1/360 & 1/360 & 0 & 0 & 0 & 0 \\
2/140 & 0.95 & 1/140 & 0 & 1/140 & 1/140 & 0 & 0 & 1/140 & 1/140 & 0 & 0 & 0 \\
2/140 & 1/140 & 0.95 & 1/140 & 0 & 1/140 & 1/140 & 0 & 0 & 0 & 1/140 & 0 & 0 \\
2/140 & 0 & 1/140 & 0.95 & 1/140 & 0 & 1/140 & 1/140 & 0 & 0 & 0 & 1/140 & 0 \\
2/140 & 1/140 & 0 & 1/140 & 0.95 & 0 & 0 & 1/140 & 1/140 & 0 & 0 & 0 & 1/140 \\
6/180 & 2/180 & 2/180 & 0 & 0 & 28/30 & 0 & 0 & 0 & 1/180 & 1/180 & 0 & 0 \\
6/180 & 0 & 2/180 & 2/180 & 0 & 0 & 28/30 & 0 & 0 & 0 & 1/180 & 1/180 & 0 \\
6/180 & 0 & 0 & 2/180 & 2/180 & 0 & 0 & 28/30 & 0 & 0 & 0 & 1/180 & 1/180 \\
6/180 & 2/180 & 0 & 0 & 2/180 & 0 & 0 & 0 & 28/30 & 1/180 & 0 & 0 & 1/180 \\
0 & 0.05 & 0 & 0 & 0 & 0.025 & 0 & 0 & 0.025 & 0.9 & 0 & 0 & 0 \\
0 & 0 & 0.05 & 0 & 0 & 0.025 & 0.025 & 0 & 0 & 0 & 0.9 & 0 & 0 \\
0 & 0 & 0 & 0.05 & 0 & 0 & 0.025 & 0.025 & 0 & 0 & 0 & 0.9 & 0 \\
0 & 0 & 0 & 0 & 0.05 & 0 & 0 & 0.025 & 0.025 & 0 & 0 & 0 & 0.9
\end{bmatrix} \tag{34}$$

The diagonal terms were chosen based on the expected sojourn time of the corresponding acceleration [4, 23]. The other terms were determined either by symmetry or by their expectation from the physical relation among models.

The true process noise covariance was set to zero and the following simple process noise covariance matrices were used: $Q^1 = (0.003)^2 I$, $Q^i = (0.008)^2 I$, $i \neq 1$, where superscript i denotes quantities pertaining to model m_i. The filter-used measurement noise covariances R^i, $\forall i$, are set equal to the true one $R = rI = 1250I$.

To have a fair comparison, all the MM algorithms used the same initial assignment of model probabilities. Each of the models m_i for $i = 1,2,3,4,5$ in topology A and for $i = 1,\ldots,9$ in topology B, has initial probability of 1/5 and 1/9, respectively.

D. Simulation Results and Discussions

Table IV lists the computational complexity in terms of relative FLOP ratios, the quality of maneuver (mode) status reports, in terms of the percent of CID, IID, and NID, average modal distances, rms mode errors, and rms position and velocity errors of the LMS, MGS, and IMM estimators. The threshold for the mode identification percentages was 0.5.

TABLE IV
CID, IID, NID Mode Identification Percentages, Relative Flop Ratios, Average Modal Distances, RMS Mode Errors, RMS Position (RMSPE) and Velocity (RMSVE) Errors, and Average Peak Position (MAXPE) and Velocity (MAXVE) Errors of LMS(λ), LMS, MGS, IMM Algorithms

	Base State Estimation Measure					Mode Identification Measure			Mode Estimation Measure	
	FLOP Ratio	RMSPE	RMSVE	MAXPE	MAXVE	CID (%)	IID (%)	NID (%)	Modal Distance	Mode Error
Random scenario for topology A:										
LMS(λ)	0.386	37.83	23.34	46.03	30.95	74.73	19.91	5.36	7.44	8.51
LMS	0.384	40.10	24.38	50.32	33.23	74.18	21.26	4.56	7.49	8.79
MGS	0.374	40.06	24.49	50.02	32.93	74.17	21.70	4.13	7.48	8.83
IMM	1	39.56	24.14	48.86	32.17	74.19	21.08	4.73	7.53	8.75
First deterministic scenario for topology A:										
LMS(λ)	0.362	35.85	19.48	73.45	66.98	75.69	20.90	3.41	8.19	7.17
LMS	0.358	37.58	19.58	73.79	66.93	75.95	20.54	3.51	8.05	7.15
MGS	0.355	37.93	19.85	74.16	67.12	75.78	20.92	3.30	8.08	7.24
IMM	1	37.15	19.42	72.88	66.63	76.06	20.31	3.63	8.08	7.14
Second deterministic scenario for topology A:										
LMS(λ)	0.423	37.86	26.05	70.11	84.65	68.12	28.38	3.50	11.63	10.94
LMS	0.382	42.43	26.53	71.61	84.33	69.54	26.70	3.76	11.29	10.72
MGS	0.388	43.21	27.17	74.61	84.40	69.05	27.38	3.57	11.39	10.90
IMM	1	41.52	25.89	70.03	84.33	70.18	25.85	3.97	11.19	10.57
Random scenario for topology B:										
LMS(λ)	0.654	35.79	21.47	43.01	30.76	74.94	15.66	9.40	7.67	8.22
LMS	0.651	36.94	22.20	44.59	30.76	74.80	17.30	7.90	7.67	8.45
MGS	0.673	36.94	22.20	44.64	30.76	74.83	17.33	7.84	7.65	8.44
IMM	1	36.93	22.20	44.52	31.01	74.76	17.26	7.98	7.69	8.44
First deterministic scenario for topology B:										
LMS(λ)	0.604	35.29	19.48	64.61	65.36	74.07	18.30	7.63	8.73	7.33
LMS	0.596	35.30	19.21	64.22	65.25	74.44	18.24	7.32	8.53	7.30
MGS	0.626	35.48	19.33	64.26	65.28	74.39	18.33	7.28	8.53	7.32
IMM	1	35.28	19.20	64.20	65.21	74.45	18.17	7.38	8.55	7.31
Second deterministic scenario for topology B:										
LMS(λ)	0.579	36.28	24.95	61.08	85.48	67.07	24.46	8.47	11.88	10.82
LMS	0.556	36.73	23.78	65.83	85.23	70.30	22.31	7.39	11.10	10.21
MGS	0.650	37.65	24.65	65.51	85.33	70.00	22.90	7.10	11.14	10.46
IMM	1	36.47	23.54	62.56	85.17	70.64	21.84	7.53	11.06	10.15

Note: LMS(λ) is LMS algorithm with forgetting (discount) factor.

It can be observed that topology B is slightly better than topology A for all MM algorithms; the LMS is slightly more cost-effective than the MGS algorithm; the LMS and MGS algorithms are much more cost-effective than the IMM estimator; the LMS estimator can handle large jumps (as in the second deterministic scenario) somewhat better than the MGS and IMM algorithms; the use of forgetting factor improves the performance of the LMS algorithm slightly, which is more significant for base state estimation in the random scenario. Note that although the LMS algorithm does depend on the underlying *digraph*, this dependence is at the same level as for other MM (e.g., the IMM) estimators. This is supported by a comparison of Topologies A and B in Table IV.

The rms position and velocity errors of the LMS, LMS(λ) (i.e., with forgetting factor), MGS, and IMM estimators are shown for topology A in Fig. 4 for the random scenario and in Fig. 5 for the first deterministic scenario. The corresponding errors for topology B are given in Fig. 6 for the first deterministic scenario. The corresponding errors for topology B for the random scenario are similar to Fig. 4, except somewhat lower, and thus are omitted. The rms position and velocity errors, along with their corresponding standard deviations of the LMS(λ) and IMM estimators are shown for topology A in Fig. 7 for the second deterministic scenario. Clearly, the LMS estimator has the smallest rms errors. Note also that although both LMS and IMM estimators have their rms errors significantly larger than the

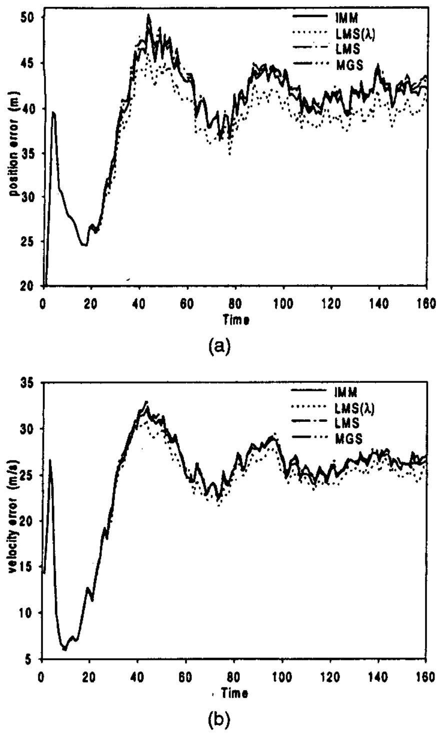


Fig. 4. RMS position and velocity errors for random scenario (topology A). (a) RMS position errors. (b) RMS velocity errors.

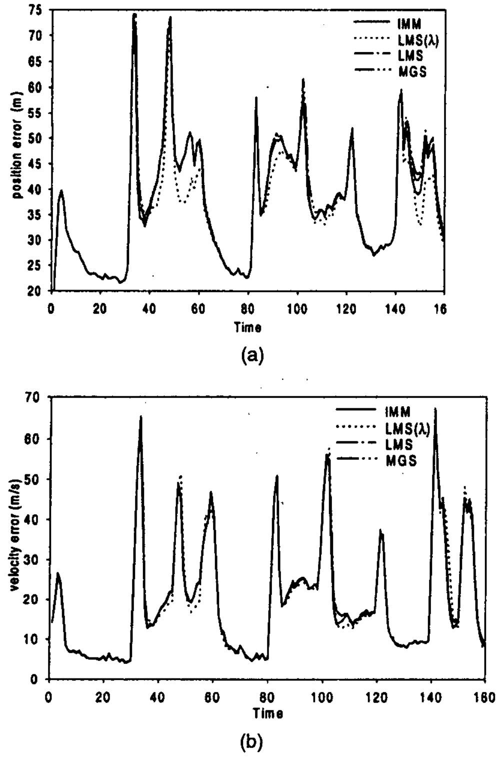


Fig. 5. RMS position and velocity errors for first deterministic scenario (topology A). (a) RMS position errors. (b) RMS velocity errors.

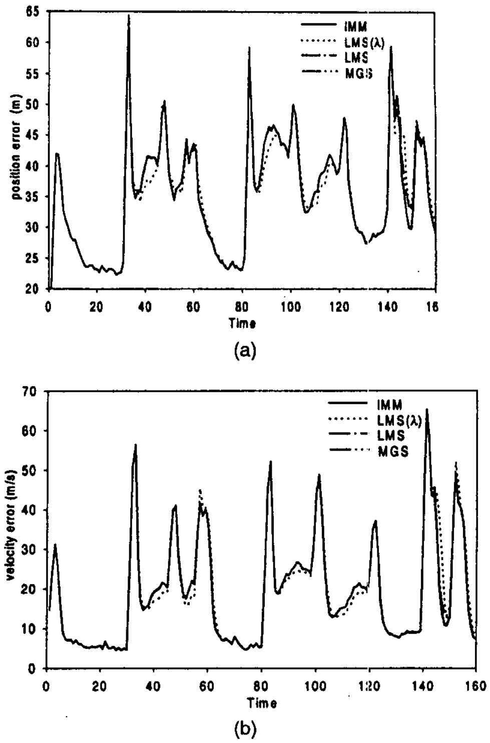


Fig. 6. RMS position and velocity errors for first deterministic scenario (topology B). (a) RMS position errors. (b) RMS velocity errors.

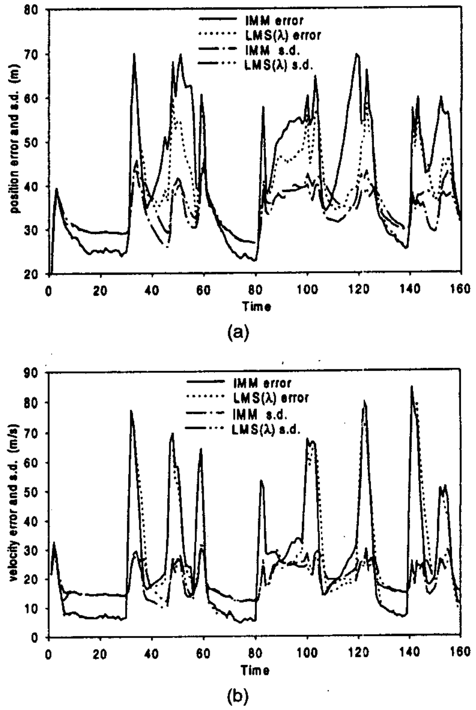


Fig. 7. RMS position and velocity errors and standard deviations for second deterministic scenario (topology A). (a) RMS position errors and standard deviations. (b) RMS velocity errors and standard deviations.

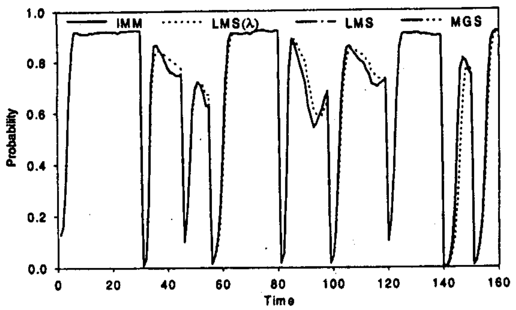


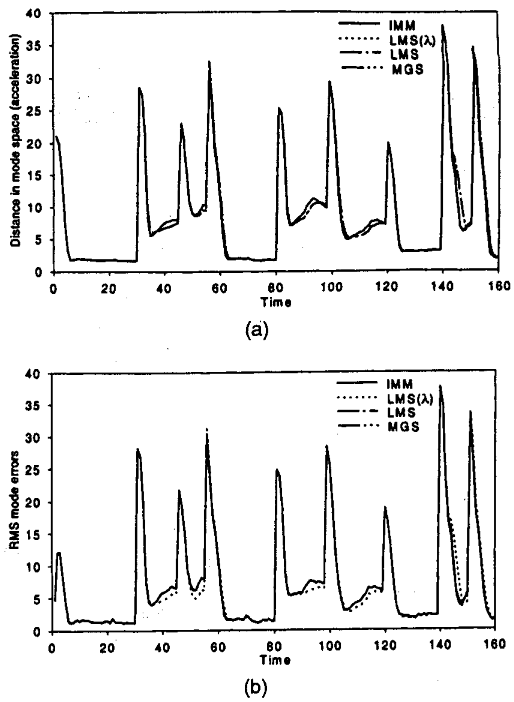
Fig. 8. Average probability of model closest to true mode for deterministic scenario.

corresponding standard deviations, especially during the mode transition transients (which is a thrust for model switching), the LMS estimator is more credible than the IMM estimator in that its error standard deviations are closer to the actual errors.

Fig. 8 shows, as calculated in the LMS, LMS(λ), MGS, and IMM algorithms for topology B, the average probability of the model that is closest to the system mode in effect at the given time; that is, $\mu_1^1,\ldots,\mu_{30}^1$, $\mu_{31}^6,\ldots,\mu_{45}^6,\ldots,\mu_{151}^1,\ldots,\mu_{160}^1$ for the first deterministic scenario. Such a plot is a compact presentation of the model probabilities when many models are involved. Clearly, all algorithms have virtually the same correct model selectability.

Fig. 9 shows for topology B, the average modal distances and the rms mode errors of the LMS, LMS(λ), MGS, and IMM algorithms for the first deterministic scenario. Similar results were obtained for topology A.

A comparison of the computational complexity between a VSMM algorithm and an FSMM algorithm depends largely on the test scenarios used and the complexity of the single-model-based filters because a VSMM estimator uses model-set adaptation and filter initialization (which are independent of the sophistication of each filter) to yield a reduction in the number of filters at any given time. Both the random and deterministic scenarios used here have more frequent mode transitions than in most practical situations. Such a scenario with frequent mode transitions is in general less favorable to a VSMM algorithm in that it increases its computation significantly while that of an FSMM algorithm will remain unchanged. Scenarios with so frequent mode transitions were designed so that many different mode transitions could be incorporated into a single mode sequence. Furthermore, the more sophisticated each model-based filter is, the more superior the LMS algorithm is to the IMM algorithm. In view of these facts, the test scenarios considered here are in favor of the IMM estimator rather than the LMS or MGS algorithm since 1) nearly-constant-velocity Kalman filters were used, which are among the simplest, and 2) the system mode undergoes frequent jumps in our test scenarios. As such, it can be reasonably concluded that the superiority of the proposed LMS algorithm will be more substantial when tested with more realistic scenarios using more sophisticated single-model-based filters.



(a)

(b)

Fig. 9. Average modal distances and rms mode errors and average modal distances for first deterministic scenario. (a) Average modal distances. (b) RMS mode errors.

VII. CONCLUSIONS

A VSMM estimator, called LMS algorithm, has been presented. It is fairly simple and easily implementable but completely general to be applicable to virtually all state estimation problems compounded with structural and/or parametric uncertainties or changes. The basic idea of the algorithm is to use only a set of models that are not unlikely. Several versions of the algorithms have been discussed. The simplest one is to delete the unlikely models based on predicted model probabilities and to activate models to which a principal model may jump so as to anticipate the possible mode transition.

A simple example of fault detection and identification and an example of tracking a maneuvering target in a 2-dimensional space have been simulated. The results demonstrate that the proposed LMS estimator is much more cost-effective than the most cost-effective FS IMM algorithm. It also outperform slightly the MGS algorithm proposed before. The ease in design and implementation is the major advantage of the LMS estimator over the MGS algorithm.

REFERENCES

[1] Ackerson, G. A., and Fu, K. S. (1970)
On state estimation in switching environments.
IEEE Transactions on Automatic Control, **AC-15** (Jan. 1970), 10–17.

[2] Anderson, B. D. O., and Moore, J. B. (1979)
Optimal Filtering.
Englewood Cliffs, NJ: Prentice-Hall, 1979.

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

[4] Bar-Shalom, Y., and Li, X. R. (1993)
Estimation and Tracking: Principles, Techniques, and Software.
Boston, MA: Artech House, 1993.

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

[6] 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** (Aug. 1988), 780–783.

[7] Brown, R. G. (1983)
Introduction to Random Siganls and Kalman Filtering.
New York: Wiley, 1983.

[8] Caputi, M. J. (1991)
Non-Gaussian estimation using a modified Gaussian sum adaptive filter.
Ph.D. dissertation, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1991.

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

[10] Daeipour, E., and Bar-Shalom, Y. (1998)
IMM tracking of maneuvering targets in the presence of glint.
IEEE Transactions on Aerospace and Electronic Systems, **34** (July 1998), 996–1003.

[11] Diesel, J. W. (1990)
Integration of GPS/INS with partitioned filters.
In C. T. Leondes (Ed.), *Analysis, Design and Synthesis Methods for Guidance and Control Systems*; AGARDOgraph 314, 1990.

[12] Drummond, O. E., Li, X. R., and He, C. (1998)
Comparison of various static multiple-model estimation algorithms.
In *Proceedings of the 1998 SPIE Conference on Signal and Data Processing of Small Targets*, **3373** (1998), 510–527.

[13] Efe, M., and Atherton, D. P. (1996)
Maneuvering target tracking using adaptive turn rate models in the interacting multiple model algorithm.
In *Proceedings of the 35th IEEE Conference on Decision and Control*, Kobe, Japan, Dec. 1996, 3151–3156.

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

[15] Jaffer, A. G., and Gupta, S. C. (1971)
On estimation of discrete processes under multiplictive and additive noise conditions.
Information Science, **3** (1971), 267.

[16] Jilkov, V. P., and Angelova, D. S. (1996)
Performance evaluation and comparison of variable structure multiple-model algorithms for tracking maneuvering radar targets.
In *Proceedings of the 26th European Microwave Conference*, Prague, Sept. 1996.

[17] Jilkov, V. P., Angelova, D. S., and Semerdjiev, T. A. (1999)
Mode-set adaptive IMM for maneuvering target tracking.
IEEE Transactions on Aerospace and Electronic Systems, **35** (Jan. 1999), 343–350.

[18] Kirubarajan, T., Bar-Shalom, Y., Pattipati, K. R., and Kadar, I. (2000)
Ground target tracking with topography-based variable structure IMM estimator.
IEEE Transactions on Aerospace and Electronic Systems, **36**, 1 (2000), 26–46.

[19] Lainiotis, D. G. (1976)
Partitioning: A unifying framework for adaptive systems, I: Estimation.
Proceedings of the IEEE, **64** (Aug. 1976), 1126–1143.

[20] Li, X. R. (1996)
Hybrid estimation techniques.
In C. T. Leondes (Ed.), *Control and Dynamic Systems: Advances in Theory and Applications*, vol. 76.
New York: Academic Press, 1996.

[21] Li, X. R. (2000)
Multiple-model estimation with variable structure—Part II: Recursive adaptive model-set approach.
To appear in *IEEE Transactions on Automatic Control*, **AC-45**, 11 (Nov. 2000).

[22] Li, X. R. (1994)
Multiple-model estimation with variable structure: Some theoretical considerations.
In *Proceedings of the 33rd IEEE Conference on Decision and Control*, Orlando, FL, Dec. 1994, 1199–1204.

[23] 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** (Sept. 1993), 186–194.

[24] 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.

[25] Li, X. R., and Bar-Shalom, Y. (1996)
Multiple-model estimation with variable structure.
IEEE Transactions on Automatic Control, **41** (Apr. 1996), 478–493.

[26] Li, X. R., and Dezert, J. (1998)
Layered multiple-model algorithm with application to tracking maneuvering and bending extended target in clutter.
In *Proceedings of the 1st International Conference on Multisource–Multisensor Information Fusion*, Las Vegas, NV, July 1998, 207–214.

[27] Li, X. R., and He, C. (1999)
Model-set choice and design for multiple-model estimation.
In *Proceedings of the 1999 SPIE Conference on Signal and Data Processing of Small Targets*, **3809**, Denver, CO (July 1999), 501–513.

[28] Li, X. R., and Zhang, Y. M. (2000)
Numerically robust implementation of multiple-model algorithms.
IEEE Transactions on Aerospace and Electronic Systems, **36**, 1 (Jan. 2000), 266–278.

[29] Li, X. R., Zhang, Y. M., and Zhi, X. R. (1999)
Multiple-model estimation with variable structure—Part IV: Design and evaluation of model-group switching algorithm.
IEEE Transactions on Aerospace and Electronic Systems, **35** (Jan. 1999), 242–254.

[30] Li, X. R., Zhi, X. R., and Zhang, Y. M. (1999)
Multiple-model estimation with variable structure—Part III: Model-group switching algorithm.
IEEE Transactions on Aerospace and Electronic Systems, **35** (Jan. 1999), 225–241.

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

[32] Magill, D. T. (1965)
Optimal adaptive estimation of sampled stochastic processes.
IEEE Transactions on Automatic Control, **AC-10** (1965), 434–439.

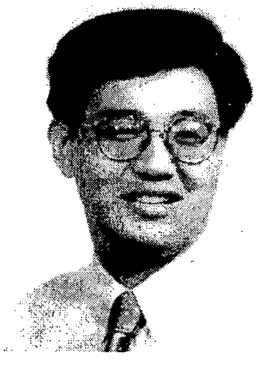
[33] Maybeck, P. S. (1982)
Stochastic Models, Estimation and Control, Vols. II, III.
New York: Academic Press, 1982.

[34] Maybeck, P. S., and Hentz, K. P. (1987)
Investigation of moving-bank multiple model adaptive algorithms.
AIAA Journal of Guidance, Control, and Dynamics, **10** (Jan.–Feb. 1987), 90–96.

[35] Munir, A., and Atherton, D. P. (1995)
Adaptive interacting multiple model algorithm for tracking a maneuvering target.
IEE Proceedings—Radar, Sonar, and Navigation, **142** (Feb. 1995), 11–17.

[36] 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.

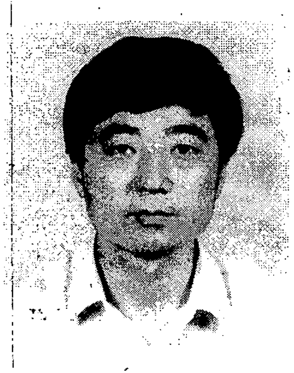
[37] Li, X. R. (2000)
Engineer's guide to variable-structure multiple-model estimation for tracking.
In Y. Bar-Shalom and W. D. Blair (eds.), *Multitarget-Multisensor Tracking: Applications and Advances,* vol. 3.
Boston: Artech House, 2000.



X. Rong Li (S'90—M'92—SM'95) received a B.S. degree and an M.S. degree from Zhejiang University, Hangzhou, Zhejiang, PRC, in 1982 and 1984, respectively, and an M.S. degree and a Ph.D. degree from the University of Connecticut, Storrs, in 1990 and 1992, respectively.

He joined the University of New Orleans in 1994. During 1986–1987 he did research on electric power at University of Calgary, Canada. He was an Assistant Professor at the University of Hartford, from 1992 to 1994. His current research interests include signal and data processing, target tracking and information fusion, stochastic systems, statistical inference, and electric power.

Dr. Li has published three books: *Multitarget-Multisensor Tracking* (YBS Publishing, 1995), *Estimation and Tracking* (Artech House, 1993), and *Probability, Random Signals, and Statistics* (CRC Press, 1999); four book chapters; more than two dozen articles in leading journals; more than 70 conference papers. Dr. Li has served as an Editor for *IEEE Transactions on Aerospace and Electronic Systems* since 1995; received a Career award and an RIA award from the National Science Foundation; received 1995 Early Career Award for excellence in Research from the University of New Orleans; served as Steering Chair and General Vice-Chair for 1998 and 1999 International Conference on Information Fusion, respectively; given numerous seminars and short courses in U.S., Europe and Asia; won several outstanding paper awards; honored as a Guest Professor by Northwestern Polytechnic University, China; listed in Marquis' *Who's Who in the World* and *Who's Who in Science and Engineering*; and consulted for several companies.



Youmin Zhang (M'99) received the B.S., M.S. and Ph.D. degrees in electrical engineering from Northwestern Polytechnical University, Xian, P. R. China, in 1983, 1986, and 1995, respectively.

Since 1986, he has been worked in the Department of Automatic Control, Northwestern Polytechnical University, Xian, P. R. China, where he was promoted as an Associate Professor in 1992 and was appointed as an Associate Chairman of the Department in 1994. During Sept. 1995–Oct. 1998, he held several research positions in the University of New Orleans, Louisiana State University, and State University of New York at Binghamton, respectively. He is currently with the Department of Electrical and Computer Engineering, The University of Western Ontario as a Postdoctoral Research Fellow. His main research interests include fault detection, diagnosis and reconfigurable control; machinery condition monitoring; signal and data processing; estimation, identification and target tracking; neural networks and applications to modeling, identification and control.

Dr. Zhang has published 30 refereed journal papers and more than 40 conference papers, edited one book and coauthored a graduate textbook. He received an award from Natural Sciences and Engineering Research Council (NSERC) of Canada for a Visiting Fellowship in Canadian Government Laboratories. He was the recipient of several research achievement awards from the National Education Commission of China, Aeronautics and Astronauts Industry Ministry of China, and Shaanxi Province Education Commission; an outstanding teaching achievement award from Shaanxi Province; several outstanding teaching and research awards from Northwestern Polytechnical University. He served as the General Chairman of the 10th Chinese Youth Conference on Automatic Control and Automation in 1994 and the international program committee member of several international conferences. He serves also as the reviewer of several refereed international journals and conferences.