Multiple-Model Estimation with Variable Structure Part IV: Design and Evaluation of Model-Group Switching Algorithm

X. RONG LI, Senior Member, IEEE
YOUMIN ZHANG
XIAORONG ZHI
University of New Orleans

A variable-structure multiple-model (VSMM) estimator, called model-group switching (MGS) algorithm, has been presented in Part III, which is the first VSMM estimator that is generally applicable to a large class of problem with hybrid (continuous and discrete) uncertainties. In this algorithm, the model-set is made adaptive by switching among a number of predetermined groups of models. It has the potential to be substantially more cost-effective than fixed-structure MM (FSMM) estimators, including the Interacting Multiple-Model (IMM) estimator. A number of issues of major importance in the application of this algorithm are investigated here, including the model-group adaptation logic and model-group design. The results of this study are implemented via a detailed design for a problem of tracking a maneuvering target using a time-varying set of models, each characterized by a representative value of the expected acceleration of the target. Simulation results are given to demonstrate the performance (based on more reasonable and complete measures than commonly used rms errors alone) and computational complexity of the MGS algorithm, relative to the fixed-structure IMM (FSIMM) estimator using all models, under carefully designed and fair random and deterministic scenarios.

Manuscript received June 17, 1997; revised January 22 and August 19, 1998.

IEEE Log No. T-AES/35/1/01502.

This work was supported in part by ONR via Grant N00014-97-1-0570, NSF via Grants ECS-9496319 and ECS-9734285, and LEQSF via Grant (1996-99)-RD-A-32.

Authors' current addresses: X. R. Li, Dept. of Electrical Engineering, University of New Orleans, New Orleans, LA 70148, E-mail: (xli@uno.edu); Youmin Zhang is on leave from the Dept. of Automatic Control, Northwestern Polytechnical University, Xian 710072, P.R. China; his current address is the Dept. of Electrical and Computer Engineering, The University of Western Ontario, London, ON N6A 5B9, Canada, E-mail: (ymzhang@overlord.engga.uwo.ca; X. Zhi, Ultratech Stepper Inc., 3050 Zanker Rd., San Jose, CA 95035.

I. INTRODUCTION

The model-group switching (MGS) algorithm presented in Part III [10] is the first variable-structure multiple-model (VSMM) estimator that is generally applicable to most state estimation problems compounded with structural and/or parametric uncertainties.

In the MGS estimator, the model-set in effect is made adaptive by switching among a number of predetermined model groups, each representing a cluster of closely related system behavior patterns or structures, according to certain "hard" rules (hard decisions). As such, only one model group has to be run at any time and thus substantial computational reduction is achieved, compared with a fixed-structure MM (FSMM) estimator based on the total model-set.

The key features of the MGS algorithm that make it very promising include the following.

1) The switching between model groups is done in two stages. The model group that is deemed likely to be a good candidate at the time is activated first without forcing the termination of the one currently in effect. The algorithm then runs their union until sufficient information is gathered to terminate one of them. This two-stage switching (i.e., activation and termination) provides a good strategy for completing the conflicting tasks of simultaneously minimizing the chance of false switching and the delay in correct switching. The only cost paid for this is an insignificant increase in computation for running the union briefly instead of a single group.

2) The initialization of the newly activated model group is done automatically due to the incorporation of the variable-structure interacting multiple-model (VSIMM) recursion of Part II [5]. This recursion assigns the probabilities to the newly activated models and initializes the filters based on these models in a natural, systematic, and near optimal manner. It also eliminates the need to design the transition probabilities for each model group.

3) The decision concerning which model group to terminate is made based on a combination of the sequential mode-set probability ratio test and the sequential model-set likelihood ratio test of Part II, which are optimal in the sense of having the quickest decision subject to prespecified lower bounds on type I and type II error probabilities.

4) The use of the optimal fusion rule of Part II in the MGS algorithm makes possible the prompt (in fact, immediate) use of the estimates from the newly activated models/filters without computational waste.

Most of these features are general. They can be and are actually incorporated into the other VSMM algorithms to be presented in the subsequent parts of this series.

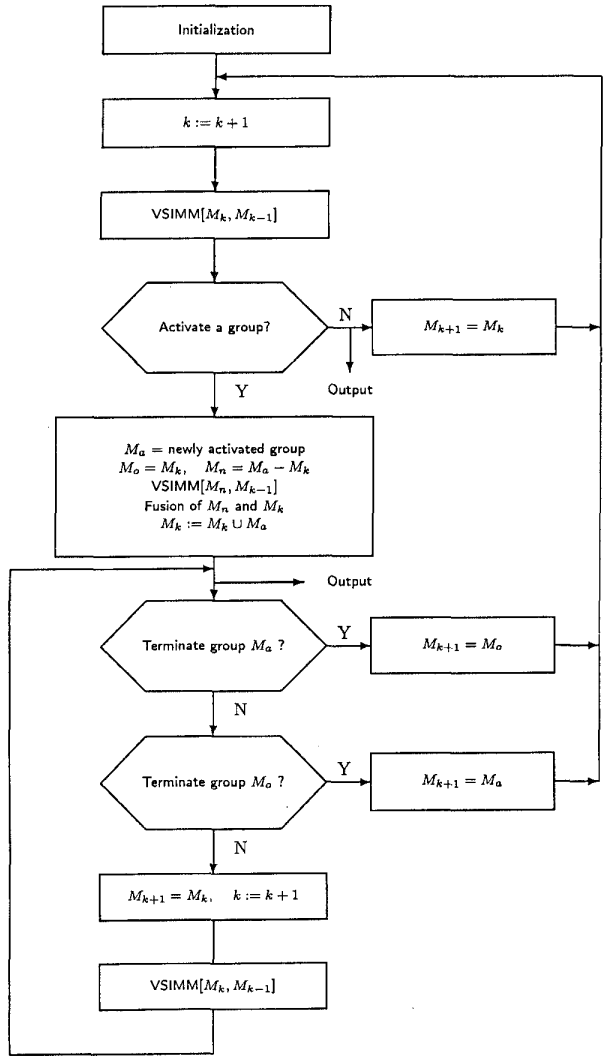


Fig. 1. Flowchart of MGS algorithm.

For details of the MGS algorithm, the reader is referred to Part III [10]. A flowchart of the MGS algorithm is found in Fig. 1.

In this paper, $\mathbf{M}_k$ stands for the (random) model-set in effect at time k; M_k for the event that the system mode at time k is in some model-set M; and the discrete time is denoted exclusively by k or κ, unless otherwise stated. For example, M_j stands for the jth model-set, rather than the model-set in effect at time j.

Although the MGS algorithm is general, its successful application requires a proper design, especially of the set of model groups and the logics for model-group activation. Such design tasks are the extra cost necessary to achieve the superiority of the MGS estimator to the FSMM estimators.

The purpose of this paper is three-fold. First, it is demonstrated how the MGS estimator should be designed, via an example of tracking a target that may undergo maneuvers. Although the design was inevitably directed to the particular application under consideration, general features of the procedure and technique are illustrated. Second, in view of the fact that the performance and computational requirements of an MM estimator, especially of a VSMM estimator, depend largely on the test scenarios, a carefully designed random scenario, along with a deterministic one, is proposed for the evaluation of MM estimators. Third, some new performance measures of MM estimators are presented based on the theoretical results of [8].

The rest of the paper is organized as follows. Section II presents a detailed design of an MGS estimator for an example of tracking a maneuvering target. The test scenarios under which the MGS algorithm is compared with the Interacting Multiple-Model (IMM) estimator are designed in Section III. Several performance measures that are useful for properly evaluating the performance and computational requirements of MM algorithms are proposed in Section IV. Simulation results are given in Section V, along with discussions. The last section provides concluding remarks.

II. DESIGN OF MODEL-GROUP SWITCHING ALGORITHM

Design of the total model-set is an important and difficult issue, common to all MM estimators. See [4] for the limited theoretical results that are available and some general discussions.

The design of the model groups should be integrated with that of the total model-set. Suppose that the total set of models has been designed. Then the design of an MGS algorithm consists of the following two major elements: 1) design of the model-set cover (i.e., model groups), and 2) design of candidate model-group activation logic and selection of the thresholds for the termination of model groups. In addition, the following design elements should also be completed, which are common to most switching MM estimators: a) selection of the model transition probability matrix, and b) tuning of the process noise covariance matrix of each filter. These design elements should be evaluated under carefully determined test scenarios.

In the sequel, each of the above elements will be discussed and our design will be presented.

A. An Illustrative Problem for Design

In our design, the following problem, adopted from [1], is considered. A virtually same problem was considered in [11, 12]. 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, [1] compared the performance of an IMM estimator with that of a Viterbi algorithm based

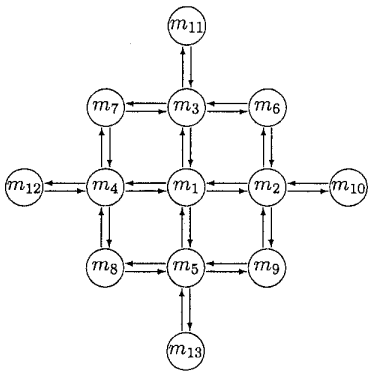


Fig. 2. Graph-theoretic representation of model-set (1)–(2).

on the following set of 12 time-invariant models, characterized by the expected acceleration vector **a**,

$$\begin{aligned} m_1 &: \mathbf{a} = [0,0]' & m_2 &: \mathbf{a} = [20,0]' \\ m_3 &: \mathbf{a} = [0,20]' & m_4 &: \mathbf{a} = [-20,0]' \\ m_5 &: \mathbf{a} = [0,-20]' & m_6 &: \mathbf{a} = [20,20]' \\ m_7 &: \mathbf{a} = [-20,20]' & m_8 &: \mathbf{a} = [-20,-20]' \\ m_9 &: \mathbf{a} = [20,-20]' & m_{10} &: \mathbf{a} = [40,0]' \\ m_{11} &: \mathbf{a} = [0,40]' & m_{12} &: \mathbf{a} = [-40,0]' \end{aligned} \tag{1}$$

These models were obtained by a quantization of the possible system mode set (space). To cover the assumed acceleration range, an additional model

$$m_{13} : \mathbf{a} = [0,-40]' \tag{2}$$

is (and should be) added to the model-set in our work. What transition relations among models (i.e., to which models a model is allowed to switch) were used in [1] is unclear since only the diagonal elements of the transition probability matrix of the assumed Markov chain for the model sequence were specified in [1]. Such ambiguity could have been avoided if a directed graph (i.e., digraph) were used to describe the topology of the model-set, as proposed in Part I [7], rather than the model-set alone.

A graph-theoretic (digraph) representation (see [7]) of these 13 models is shown in Fig. 2, 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). The topology shown in this figure will be used in our design. Fig. 2 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. 2 is exactly proportional to the actual distance between the two models in their acceleration space.

Note that for simplicity a model is allowed to switch to its nearest neighbor(s) only. Better results could be obtained if other types of model switching are allowed, such as those between second nearest neighbors (e.g., m_2 and m_3, and m_6 and m_{10}) (see Part V [9]).

Please keep in mind that the application of an MM estimator based on the above models, characterized by the constant expected acceleration along x and y directions, is not necessarily most effective for tracking a target that may undergo a turning motion [3]. For example, such a maneuver usually has an almost constant *lateral* acceleration normal to the velocity vector, rather than a constant acceleration in x and y directions. Nevertheless, the above models were adopted here to demonstrate how an MGS algorithm can be designed and to compare its performance and computational requirements with those of the corresponding fixed-structure estimator. In fact, the above vector **a** may actually be (treated as) a parameter other than the expected acceleration that characterizes a maneuver and the above model-set is a quantization of the corresponding parameter space. Then the MGS estimator developed and designed here may be effectively applied to some other problems, such as fault detection and isolation.

B. Model-Groups (Cover) Design

Model-group design is closely related to the design of the total model-set and its topology (i.e., graphical relation), both of which are highly dependent on the particular application considered. Given the topology of the total model-set, a set of models should be grouped together only if the system modes (behavior patterns or structures) represented by these models are "closest" in some sense (except possibly for a "nominal" or initial group). This is, however, only a necessary condition. The model-group design should also consider the following aspects.

1) It should provide a mechanism for switching from one group to another if a corresponding mode transition is not with an extremely low probability in reality. It is because of this that model groups should have common models.

2) It should have as few common models among groups as possible to reduce computation.

3) It should prevent frequent switching among model groups in most scenarios for the application under consideration.

4) It should not make difficult the design and implementation of a proper and uncomplicated logic for activating a model group.

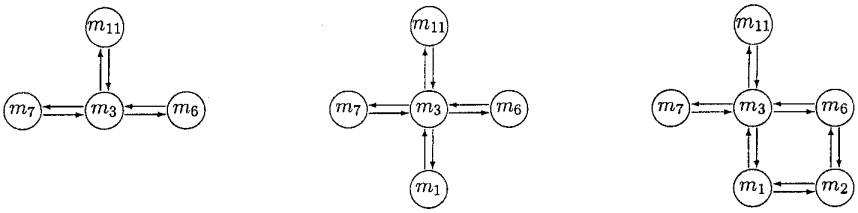


Fig. 3. Three representative members of three candidate covers of Fig. 2.

5) It should facilitate a proper assignment of the initial probabilities to the newly activated models and initialization of the filters based on these models.

These requirements are conflicting. For example, a cover consisting of large groups of models will have on the average a small probability of group switching but high computational complexity within each model group. Also, a larger overlap between two model groups will make it easier to design a better group activation logic and a better assignment of the initial model probabilities.

For example, three candidate covers of the total model-set given in Fig. 2 are

$$C_1 = \{M_1, M_2^1, M_3^1, M_4^1, M_5^1\}$$

$$C_2 = \{M_1, M_2^2, M_3^2, M_4^2, M_5^2\} \tag{3}$$

$$C_3 = \{M_1, M_2^3, M_3^3, M_4^3, M_5^3\} \tag{4}$$

where

$$M_1 = \{m_1, m_2, m_3, m_4, m_5\} \tag{5}$$

$$M_2^1 = \{m_2, m_{10}, m_6, m_9\}$$

$$M_3^1 = \{m_3, m_{11}, m_7, m_6\}$$

$$M_4^1 = \{m_4, m_{12}, m_8, m_7\}$$

$$M_5^1 = \{m_5, m_{13}, m_9, m_8\}$$

$$M_2^2 = \{m_2, m_{10}, m_6, m_1, m_9\}$$

$$M_3^2 = \{m_3, m_{11}, m_7, m_1, m_6\}$$

$$M_4^2 = \{m_4, m_{12}, m_8, m_1, m_7\}$$

$$M_5^2 = \{m_5, m_{13}, m_9, m_1, m_8\}$$

$$M_2^3 = \{m_2, m_{10}, m_6, m_1, m_9, m_5\}$$

$$M_3^3 = \{m_3, m_{11}, m_7, m_1, m_6, m_2\}$$

$$M_4^3 = \{m_4, m_{12}, m_8, m_1, m_7, m_3\}$$

$$M_5^3 = \{m_5, m_{13}, m_9, m_1, m_8, m_4\}. \tag{6}$$

A systematic labeling and ordering of the models in a group, such as given above, will facilitate the implementation of the MGS algorithm (and probably most other VSMM algorithms).

Clearly, in the context of maneuvering target tracking, M_1 in effect means that very likely the target is not maneuvering and M_2, M_3, M_4, and M_5 represent different maneuvers. For fault detection and isolation, M_1 represents the "nominal" mode while M_2, M_3, M_4, and M_5 correspond to different fault modes. In all the three cover designs, M_1 is the "nominal" group, which represents the nominal system mode with a proper mechanism for switching to nonnominal groups.

This is a typical example of a multidimensional problem, which usually requires a large number of models.

Fig. 3 shows one typical member model group from each of the three candidate covers. There are of course other covers. Using the above criteria, cover C_1 does not seem a good one since it does not provide a good mechanism for switching from M_i^1, $i \neq 1$, to M_1 and the switching among groups may be too frequent. Cover C_2 may have trouble deciding which model group (e.g., M_2^2 or M_5^2) to use if the true system mode has an almost equal acceleration in magnitude in both x and y directions (e.g., $\mathbf{a} = [10, -10]'$), in which cases the union of two model groups (e.g., $M_2^2 \cup M_5^2$) may have to be run in the MGS algorithm. This is a case in which decomposition of model-group switching into model-group activation and termination is clearly beneficial. Cover C_3 overcomes these problems at the cost of a higher computational burden. Overall it is unclear whether C_2 is superior to C_3. For example, cover C_3 will have the smallest probability of model-group switching but the highest computation for each model group. Since the only apparent problem with C_2 is for cases in which $|a_x| \approx |a_y|$, which is not a serious one since the proposed MGS algorithm will take care of this problem by running the union of the corresponding two model groups, all design results presented here are based on cover C_2.

C. Design of Model-Group Activation Logic

Similarly to the design of any hard decision rule, the design of the logic for the activation of the model group has to satisfy the following requirements:

1) The average decision delay should be as small as possible.

2) The probability of a false decision should be as low as possible.

3) The logic should be as simple and computationally efficient as possible.

These requirements are conflicting and it seems that no single optimality criterion is entirely reasonable. As a result, there is no truly optimal design of such a logic. Generally speaking, a good design can only be obtained by starting from insights into the problem (in particular the application under consideration) and finding a good compromise among the above requirements via a trial-and-error procedure. In doing so, the following general guidelines are useful.

1) The average delay of a correct decision affects mostly the peak estimation errors during the transition of the system mode from one within a model group to another mode that is best represented (covered) by another model group. The greater the average delay is, the higher the peak errors are and the longer the large errors last. Note, however, that it is possible to further reduce the peak errors in many cases by some ad hoc techniques, such as tuning of process noise covariance, adjustment of the error covariances (by say, a fudge or forgetting factor), and modification of the original algorithm (e.g., the instantaneous feeding of [6]). These adjustments, however, were not used to obtain the results presented here.

2) A false decision has its major impact on the stationary (steady-state) estimation errors. A significantly higher probability of false decision will lead to a substantial increase in the steady-state errors.

3) The simplicity of the logic affects primarily the central processing unit (CPU) time and has a less significant effect on the floating point operation (flop) count.

Based on the above considerations and the general discussion of Subsection IIIC1 of Part III and after a brief trial-and-error adjustment, the following simple activation logic was adopted in our design.

1) Activate group M_j while $\mathbf{M}_k = M_1$, where $j \neq 1$, if *both* of the following two conditions are satisfied

$$\text{likelihood condition:} \quad L_k^j = \max_{m_l \in M_k} L_k^l$$

$$\text{probability condition:} \quad \mu_k^j > t_0$$

where μ^j and L^j are the probability and likelihood of model m_j, respectively. For example, if model m_2 has the highest likelihood and its probability exceeds the threshold t_1 while group M_1 is in effect, then group M_2 is activated.

2) Activate group M_1 while $\mathbf{M}_k = M_j$, where $j \neq 1$, if *both* of the following two conditions are satisfied

$$\text{likelihood condition:} \quad L_k^1 = \max_{m_l \in M_k} L_k^l$$

$$\text{probability condition:} \quad \mu_k^1 > t_1.$$

For example, if model m_1 has the highest likelihood and its probability exceeds the threshold t_1 while group M_2 is in effect, then group M_1 is activated.

3) Activate group M_j while $\mathbf{M}_k = M_i$, where $i, j \neq 1$, if *both* of the following two conditions are satisfied for some $m_n \in M_j$, $m_n \neq m_1$,

$$\text{likelihood condition:} \quad L_k^n = \max_{m_l \in M_i} L_k^l$$

$$\text{probability condition:} \quad \mu_k^n > t_2.$$

For example, if model m_6 has the highest likelihood and its probability exceeds the threshold t_2 while group M_2 is in effect, then group M_3 is activated.

4) Direct switching between M_2 and M_4, or M_3 and M_5 is not allowed. Such transition may go through M_1 as an intermediate group.

In our design, $t_0 = t_1 = t_2 = 0.3$ were used for simplicity.

Other activation decision rules are possible. Some theoretical results were presented in Part II [5] for model-set activation. They should lead to better performance, however, at a higher computational cost. This is studied later.

D. Selection of Group Termination Thresholds

The design of the candidate model-group (activation) logic is best integrated with the selection of the thresholds for the termination of model groups.

In most cases, the threshold pairs $\{t_1^\mu, t_2^\mu\}$ and $\{t_1^L, t_2^L\}$ should be selected to yield a low probability of running the union of the model groups. In some other cases, if computational resources permit, they may be chosen such that the above probability is not low. Our experience indicates that the performance and computation of the MGS algorithm are not sensitive to the selection of these thresholds provided that the system mode does not stay in the central area of the union of two model groups for a long time.

It is our experience that after the candidate model group is activated, the ratio of its probability (or likelihood) to that of the current model group varies following a definite pattern: it either increases or decreases dramatically, depending on whether the candidate group is the right one or not. Thus, it is quite easy to determine the thresholds by observing first the average ratio data.

Although in general, different sets of thresholds $\{t_1^\mu, t_1^L, t_2^\mu, t_2^L\}$ may be needed for different pairs of model groups to yield the best performance-to-computation ratio, our experience indicates that a single set of thresholds usually works well enough because the probability and/or likelihood ratios increase or decrease rapidly enough. If, however, the core models of the competing group are deleted from the model groups, as shown by (8)–(9) of Part III, then different threshold sets may have to be used. Note that for this example, if M_1 is the current model group M_o and M_3 is the candidate model group M_a, then

$$M_a - C_o = M_1 - C_3 = \{m_3, m_{11}, m_7, m_6\}$$

$$M_o - C_a = M_3 - C_1 = \{m_1, m_2, m_4, m_5\}$$

where C_o and C_a are the (sets of) core models of the model groups M_o and M_a, respectively. Thus, better results should be obtained if (8)–(9) of Part III were used. If, however, M_2 is the current model group M_o and M_3 is the candidate model group M_a, then

$$M_a - C_o = M_2 - C_3 = M_2$$

$$M_o - C_a = M_3 - C_2 = M_3.$$

That is, (8)–(9) of Part III reduces to the ratios of model-set likelihoods and probabilities.

In our design, the core models were not deleted and the following values were used for simplicity:

$$t_1^{\mu} = t_1^{L} = 0.95, \qquad t_2^{\mu} = 1.1, \qquad t_2^{L} = 1.$$

E. Other Design Elements

The following transition probability matrix was used in our examples given in the next section:

$$\Pi = \begin{bmatrix} 116/120 & 1/120 & 1/120 & 1/120 & 1/120 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.02 & 0.95 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0.01 & 0.01 & 0 & 0 & 0 \\ 0.02 & 0 & 0.95 & 0 & 0 & 0.01 & 0.01 & 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0.02 & 0 & 0 & 0.95 & 0 & 0 & 0.01 & 0.01 & 0 & 0 & 0 & 0.01 & 0 \\ 0.02 & 0 & 0 & 0 & 0.95 & 0 & 0 & 0.01 & 0.01 & 0 & 0 & 0 & 0.01 \\ 0 & 1/30 & 1/30 & 0 & 0 & 28/30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/30 & 1/30 & 0 & 0 & 28/30 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/30 & 1/30 & 0 & 0 & 28/30 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/30 & 0 & 0 & 1/30 & 0 & 0 & 0 & 28/30 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9 \end{bmatrix}. \quad (7)$$

The diagonal terms were chosen based on the expected sojourn time of the corresponding acceleration [2, 6]. The other terms were determined either by symmetry or their expectation from the physical relation among models. Equation (7) was used in both the MGS and the fixed-structure IMM (FSIMM) algorithms.

Note that if the VSIMM cycle of Part II is used, then there is no need to design the transition probability matrix for each model group 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 group. All these are done naturally and systematically. See Part II [5] for details.

The true process noise covariance was set to zero and the following simple process noise covariance matrices were used in our work: $Q^1 = (0.003)^2 I$, $Q^i = (0.008)^2 I$, $i \neq 1$. The filter used measurement noise covariances R^i, $\forall i$, are set equal to the true $R = 1250I$.

The MGS algorithm was initialized by the model group M_1 with evenly distributed model probabilities. To have a fair comparison, the FSIMM estimator was initialized with the same initial conditions as those of the MGS algorithm, including $\mu_0^i = 1/5$ for $i = 1,\ldots,5$ and $\mu_0^i = 0$ for $i = 6,\ldots,13$. Note that were the IMM estimator initialized with evenly distributed model probabilities, poorer results would be obtained for the random and deterministic scenarios considered here.

III. DESIGN OF TEST SCENARIOS

It should be emphasized that a comparison between an FSMM algorithm and a VSMM algorithm depends to a large degree on the scenarios used for the evaluation.

Both deterministic and random scenarios were designed and used for testing and comparing the performance and computational requirements of the proposed MGS algorithm with the FSIMM estimator using all 13 models.

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. The amplitude a and angle θ are both 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.

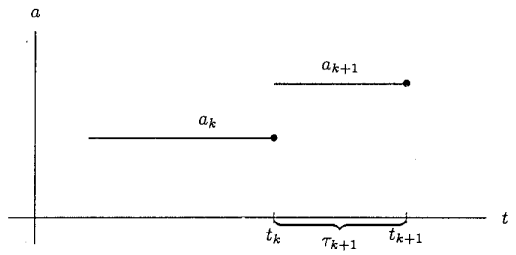


Fig. 4. Semi-Markov process model for acceleration sequence.

Referring to Fig. 4, the following model for **a** is proposed.

1) Denoting by τ_k the sojourn time for the state $a = a_k$, its conditional probability density is

$$p(\tau_k \mid a_k) = \frac{1}{c}\mathcal{N}(\tau_k;\overline{\tau},\sigma_\tau^2), \qquad \tau_k > 0 \tag{8}$$

where $\mathcal{N}(x;\bar{x},\sigma^2)$ stands for Gaussian (normal) probability density function of x with mean $\bar{x}$ and variance σ^2; c is a normalization constant; and $\overline{\tau}$ and σ_τ are two constants, depending on a_k. In plain terms, the random sojourn time is assumed to be truncated Gaussian which is state dependent.

2) Denoting by a_{k+1} the magnitude of acceleration over the period $(t_k,t_{k+1}]$, its conditional probability densities are

$$p(a_{k+1} \mid a_k) = P_0\,\delta(a_{k+1}) + P_M\,\delta(a_{k+1} - a_{\max}) + [1 - P_0 - P_M]\frac{1}{a_{\max}}1[a_{k+1};(0,a_{\max})] \tag{9}$$

where P_0 and P_M are in general functions of a_k; δ is the delta function; and $1(x;R)$ is the indicator function, defined by

$$1(x;R) = \begin{cases} 1 & x \in R \\ 0 & x \notin R \end{cases}. \tag{10}$$

In other words, the acceleration magnitude has probability masses of P_0 and P_M to be zero and maximum, respectively, and is uniform over the values in between.

3) Denoting by θ_{k+1} the angle of acceleration over the period $(t_k,t_{k+1}]$, its conditional probability density is

$$p(\theta_{k+1} \mid a_k = 0) = \frac{1}{2\pi}1(\theta_{k+1};(-\pi,\pi]) \tag{11}$$

i.e., θ_{k+1} (conditioned on $a_k = 0$) is uniformly distributed over $(-\pi,\pi]$, and

$$p(\theta_{k+1} \mid a_k > 0) = \mathcal{N}(\theta_{k+1};\theta_k,\sigma_\theta^2). \tag{12}$$

Equation (11) is reasonable since every angle is equally likely. Equation (12) is assumed in view of the fact that the angles of the consecutive accelerations are not necessarily equal but usually coupled.

The following parameters were used in our design:

$$\overline{\tau} = \overline{\tau}_M + \frac{a_{\max} - a}{a_{\max}}(\overline{\tau}_0 - \overline{\tau}_M), \tag{13}$$

$$\sigma_\tau = \frac{1}{12}\overline{\tau}_a, \qquad \overline{\tau}_M = 10, \qquad \overline{\tau}_0 = 30$$

$$P_M = 0.1, \qquad a_{\max} = 37, \qquad \sigma_\theta = \pi/12, \tag{14}$$

$$P_0 = \begin{cases} 0.6 & a_k \neq a_{\max} \\ 0.8 & a_k = a_{\max} \end{cases}.$$

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

A random scenario similar to the above is also proposed for testing fault detection and identification algorithms in [13].

The use of such a random test scenario alleviates the dependence of the MM algorithm performance on various artifects of a scenario. With such a random scenario, it is difficult, if not virtually impossible, to design a VSMM algorithm with subtle tricks that are effective only for certain scenarios. Of course, it cannot prevent intentional cheat.

The most serious drawback of such a random scenario is that the peak estimation errors and the steady-state errors are no longer separable when the results are averaged over multiple Monte Carlo runs. For 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 random scenario and the following deterministic scenario are included here. The system mode sequence, characterized by the sequence of accelerations $\mathbf{a}_k$, is specified by

$$\mathbf{a}_k = \begin{cases} [0,0]' & 1 \le k \le 30 \\ [18,22]' & 31 \le k \le 45 \\ [2,37]' & 46 \le k \le 55 \\ [0,0]' & 56 \le k \le 80 \\ [25,2]' & 81 \le k \le 98 \\ [-2,19]' & 99 \le k \le 119 \\ [0,-1]' & 120 \le k \le 139 \\ [38,-1]' & 140 \le k \le 150 \\ [0,0]' & 151 \le k \le 160 \end{cases}. \tag{15}$$

Note that both the random and deterministic scenarios designed here have more frequent mode transitions than in most practical situations. This is less favorable to a VSMM algorithm, including the MGS algorithm, than to an FSMM estimator. This is explained later. Scenario (15) with so frequent mode transitions was designed so that many different mode transitions are incorporated into a single mode sequence.

IV. PERFORMANCE MEASURES OF MULTIPLE-MODEL ESTIMATORS

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 [8]. Other components may include robustness, parallelism, implementability, etc.

In this work, the following additional measures of the mode identification capability based on the theoretical results of [8] are used.

1) The first additional measure is the percentages of correct mode identification (CID), incorrect mode identification (IID), and no mode identification (NID).

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

b) An incorrect identification (IID) is obtained if the model with the highest probability that exceeds the threshold is not the one 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.

This measure is extended to include percentages of missed fault detection and false alarm in [13].

2) The average distances between the system mode in effect at a given time k and the models used in the MM algorithm at k, given by

$$\text{average } \textit{modal distance} \text{ at } k = \frac{1}{N}\sum_{n=1}^{N}\sum_{m_i \in M_k} \|s_k - m_i\| P\{s_k = m_i \mid s_k \in M_k, z^k\} \tag{16}$$

where N is the number of Monte Carlo runs and $\|s_k - m_i\|$ in the current example is simply the distance in the acceleration space (i.e., the distance between the true acceleration and the acceleration assumed in model m_i).

3) The rms error of mode estimation, defined by

$$\text{rms } \textit{mode error} \text{ at } k = \sqrt{\frac{1}{N}\sum_{n=1}^{N} \|s_k - \hat{s}_{k|k}\|^2} \tag{17}$$

where the mode estimate is defined by

$$\hat{s}_{k|k} = \sum_{m_i \in M_k} m_i P\{s_k = m_i \mid s_k \in M_k, z^k\} \tag{18}$$

and the average rms error is defined by

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

In the above s_k, $\hat{s}_{k|k}$, and m_i could be vector-valued and thus if Euclidean norm is used, then

$$\|s_k - \hat{s}_{k|k}\|^2 = (s_k - \hat{s}_{k|k})'(s_k - \hat{s}_{k|k}) \tag{19}$$

and similarly for $\|s_k - m_i\|$.

Note the difference and similarity between the modal distance and mode error. For our example, the mode space is simply the acceleration space. Thus, the modal distance is the weighted sum of the distances between the true acceleration and the acceleration vectors used in the models at the time, while the rms mode error is the rms deviation of the weighted sum of the accelerations assumed in the models from the true acceleration.

Use of the CID, IID, and NID for the comparison of MM estimators with different total model-sets in general does not make sense. In fact, misleading results may be obtained. For example, it is almost always the case that a 2-model MM estimator has better CID, IID, and NID than a 100-model MM estimator for the same problem because these percentages do not consider how fine the mode space is quantized by the model-set. The use of the average modal distance or rms mode error does not have this limitation. It is, however, less revealing and less handy. Also, for some practical problems, it may be hard to define such a distance or error if different models are characterized by distinct quantities, rather than different values of the same quantity.

V. SIMULATION RESULTS AND DISCUSSIONS

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

It was found that the use of (second-order) nearly constant velocity (CV) models [2] with specified expected accelerations led to significantly better results than (third-order) nearly constant acceleration (CA) models [2] for both MGS and IMM algorithms. Also, poor results would be obtained if the accelerations of a model were taken as part of the state. This is due to the fact that the accelerations of all models in the group would be driven by the measurement sequence from their specified (expected) values towards the true system mode and thus the model-set (group) would no longer be able to provide a coverage wide enough to handle effectively sudden jumps in the system mode that may occur later. Therefore, the results reported here are based on CV models with specified accelerations.

The rms position errors, velocity errors, and their corresponding standard deviations of the MGS

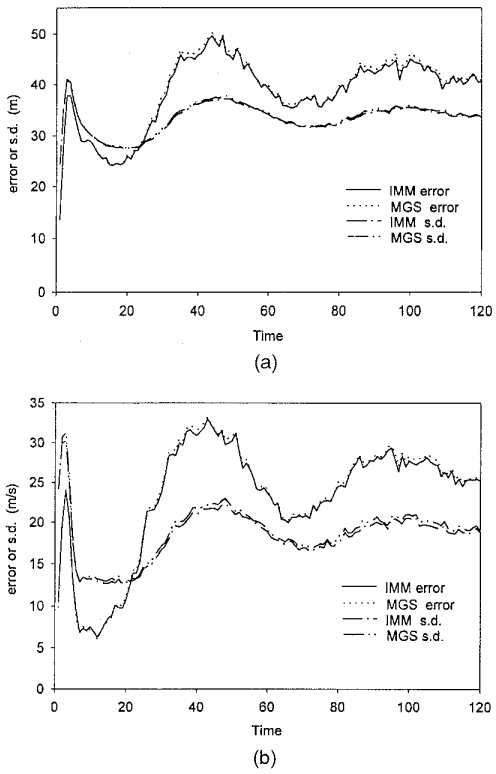


Fig. 5. (a) RMS position errors and standard deviations for random scenario. (b) RMS velocity errors and standard deviations for random scenario.

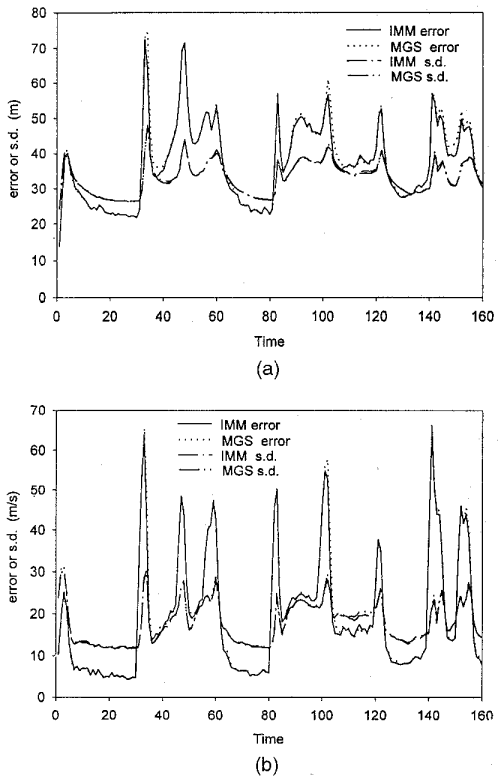


Fig. 6. (a) RMS position errors and standard deviations for deterministic scenario. (b) RMS velocity errors and standard deviations for deterministic scenario.

estimator and the FSIMM estimator that uses all 13 models are shown in Fig. 5 for the random scenario and in Fig. 6 for the deterministic scenario. Note that both algorithms have their rms errors significantly higher than their corresponding standard deviations during the mode transition transients, which is the thrust for model switching. Since the acceleration is specified in m/s^2, the results are in meters and seconds (the sampling interval $T = 1$ s). In fact, the same results hold for any physical dimension, such as feet, depending on the dimension used for the measurement z and the Q and R matrices.

Figs. 7 and 8 show the average modal distances, defined by (16), and the rms mode errors, defined by (17), where $\|s_k - \hat{s}_{k|k}\|$ is actually the acceleration estimation error (i.e., the difference between the true and estimated accelerations) obtained in nth run,

$$\|s_k - \hat{s}_{k|k}\|^2 = (a_k^1 - \hat{a}_k^1)^2 + (a_k^2 - \hat{a}_k^2)^2 \qquad (20)$$

where

$$\hat{a}_k = \begin{bmatrix} \hat{a}_k^1 \\ \hat{a}_k^2 \end{bmatrix} = \sum_{m_i \in M_k} a^i P\{s_k = m_i \mid s_k \in M_k, z^k\}$$

and a^i is the acceleration vector assumed in model m_i. Note also that each model used is a nearly constant

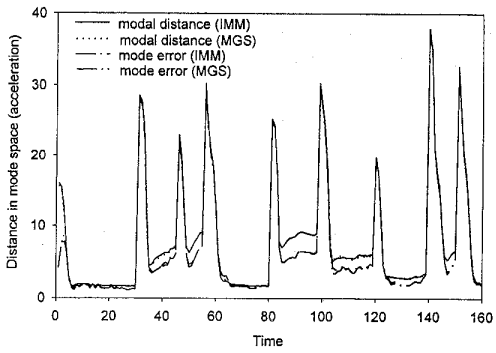


Fig. 7. RMS mode errors and average modal distances for deterministic scenario.

velocity one with a specified acceleration and thus the acceleration is not part of the state.

Fig. 9 shows for the deterministic scenario, the probabilities of the models that are closest to the system mode in effect at the given time, given by, for the scenario (15),

$$\mu_1^1, \ldots, \mu_{30}^1, \mu_{31}^6, \ldots, \mu_{45}^6, \ldots, \mu_{151}^1, \ldots, \mu_{160}^1.$$

It provides a compact way of presenting the model probabilities when many models are used in the MM estimator.

TABLE I
Percentages of Correct (CID), Incorrect (IID), No (NID) Mode Identification, Relative Flop Ratio, Average Modal Distance, RMS Mode Errors, Average RMS Position (RMSPE) and Velocity (RMSVE) Errors of MGS and IMM Algorithms

	flops	CID (%)	IID (%)	NID (%)	Modal Distance	Mode Error	RMSPE	RMSVE
Random scenario:								
MGS	0.366	75.32	20.19	4.49	7.25	8.50	38.98	17.55
IMM	1	75.24	19.59	5.17	7.31	8.45	38.50	17.33
Deterministic scenario:								
MGS	0.356	75.97	20.72	3.31	8.05	7.17	37.84	19.68
IMM	1	76.24	20.15	3.61	8.05	7.09	37.11	19.32

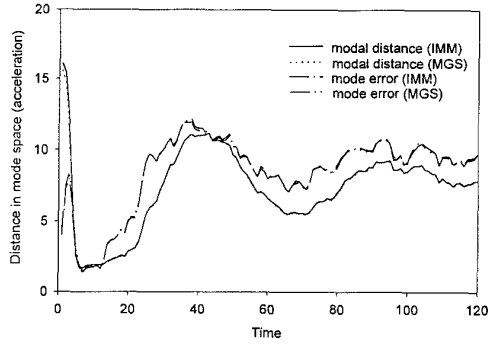


Fig. 8. RMS mode errors and average modal distances for random scenario.

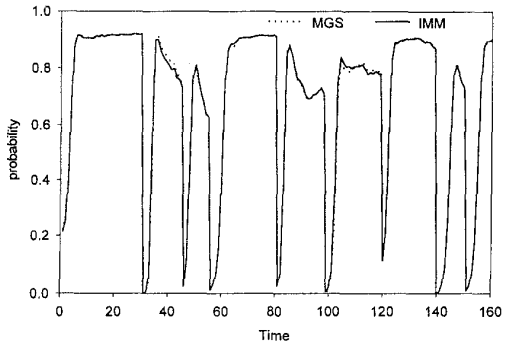


Fig. 9. Average probabilities of model closest to true mode for deterministic scenario.

Table I lists the computational complexity in terms of relative FLOP ratios, the quality of maneuver (mode) status reports, in terms of the percentages of CID, IID, and NID, average modal distances, rms mode errors, and rms position and velocity errors of the MGS and IMM estimators. 0.5 was used as the threshold for the mode identification percentages. The CPU time used for the two algorithms could not be reported because of a bug in the MATLAB "cputime" function (it may report negative time).

Note that our design is fairly general (not quite specific to the problem considered and no ad hoc tricks were used), except for the candidate activation logic, because our purpose is to compare the general MGS algorithm with the IMM algorithm. Better results can be obtained for the MGS algorithm by fine tuning the design according to the specifics of the problem.

It should be emphasized that a comparison of the computational complexity between a VSMM algorithm and an FSMM algorithm depends to a large degree on the test scenarios used and the complexity of the single-model-based filters. This is clear for the MGS algorithm since it operates fewer filters at any given time than the FSMM algorithm at the cost of computing group adaptation logic and filter initialization (which are virtually independent of how sophisticated the model-based filters are). A scenario with more frequent mode transitions, as those used here, will increase the computation of the VSMM algorithm significantly while that of an FSMM algorithm will remain unchanged. Furthermore, the more sophisticated each model-based filter is, the more superior the MGS 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 MGS algorithm since 1) second-order (nearly constant velocity) Kalman filters were used, which are among the simplest; 2) the system mode undergoes frequent jumps in our test scenarios. As such, it can be reasonably concluded that *the superiority of the proposed MGS algorithm will be more substantial when tested with more realistic scenarios using more sophisticated single-model-based filters.*

Next, assume the target being tracked may be a fighter aircraft that may have a piecewise constant acceleration up to $8g$ (80 m/s^2) in any direction. Suppose that the same 13 models of (1)–(2) are used, except that the expected accelerations are scaled up by a factor 2. Consider the following MM estimators.

IMM13: Uses all 13 models of Fig. 2 with transition probability matrix of (7).

MGS: Uses all the 13 models of Fig. 2 with Cover C_2, defined by (3), and transition probability matrix of (7).

IMM5: Uses the 5 models of Fig. 10(a).

IMM9: Uses the 9 models of Fig. 10(b).

TABLE II

Percentages of Correct (CID), Incorrect (IID), No (NID) Mode Identification, Relative FLOP Ratio, Average Modal Distances, RMS Mode Errors, Average RMS Position (RMSPE) and Velocity (RMSVE) Errors of MGS and IMM Algorithms

	flops	CID (%)	IID (%)	NID (%)	Modal Distance	Mode Error	RMSPE	RMSVE
MGS	0.392	72.87	23.84	3.29	15.60	18.34	50.48	41.55
IMM13	1	72.83	23.56	3.61	15.65	18.25	49.86	41.16
IMM9	0.572	77.67	17.15	5.18	18.68	18.82	58.21	38.53
IMM5	0.250	80.53	18.15	1.32	23.46	26.23	55.60	52.73

Note: Warning: Do not compare CID, IID, and NID for algorithms with different total model-sets.

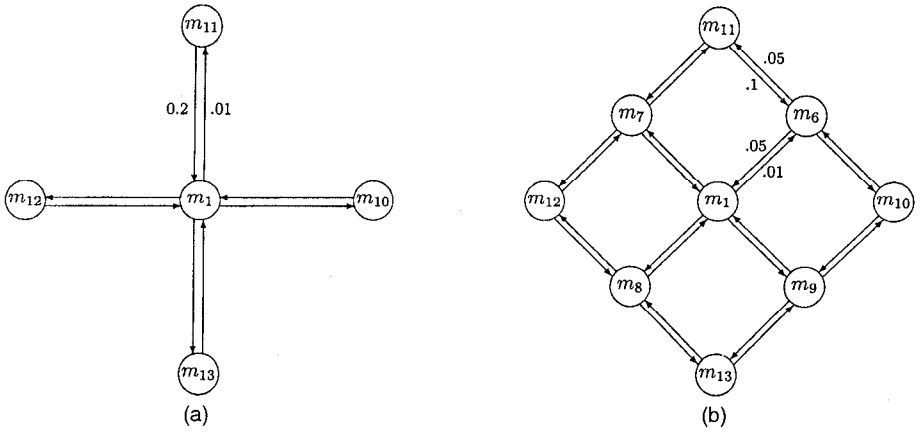


Fig. 10. Digraph assumed in IMM5 and IMM9 algorithms.

In Fig. 10, all the other transition probabilities are given by symmetry and the self-transition probabilities may be obtained from 1 minus the sum of all transition probabilities from a model. Note that all models are allowed to switch only to its nearest neighbors directly.

Consider the semi-Markov random scenario of Section 3 with a maximum acceleration $a_{\max} = 75$ and $P_0 = P_M = 0.3$.

This example demonstrates the benefit of using a large number of models and the performance superiority of the MGS estimator to the FSIMM estimator with comparable computational requirements.

Note that as explained in the previous section, a comparison of the CID, IID, and NID of MM estimators with different total model-sets is misleading, as in Table II. The average modal distance and rms mode error are better in this case.

Fig. 11 shows, for the second random scenario, the rms position errors of the MM estimators.

In fact, comparisons between the MGS and IMM algorithms were also conducted for several other deterministic scenarios in which the mode transitions occurred less frequently and/or with a smaller jump in magnitude (i.e., with a smaller jump distance in the mode space). The results showed additional improvement of the MGS algorithm on the IMM algorithm. Due to space limitation, these results are not presented here.

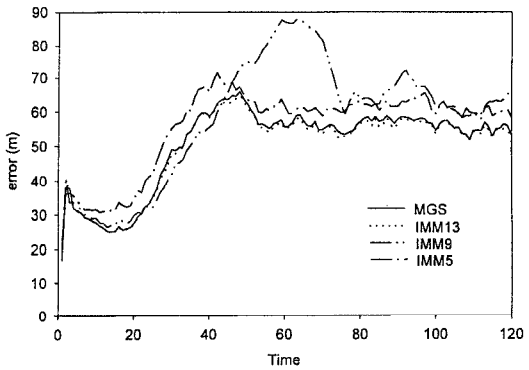


Fig. 11. RMS position errors for second random scenario.

From our simulation results, including those not presented here, it is clear that the MGS algorithm has computational complexity only about one-third of that of the IMM estimator while they have essentially identical performance. If the total model-set is larger, then this computational reduction can be expected to be more substantial. On the other hand, if the same computational constraints are imposed on the MGS and IMM algorithms, then the MGS estimator will outperform the IMM algorithm significantly.

It should be pointed out that the actual mode identification quality of the algorithms is more or less better than those listed in Tables I and II, especially for the random scenario, because the system mode may locate somewhere which has an almost equal distance to more than one model.

It should also be pointed out that although only jump linear systems are considered for simplicity, the MGS algorithm is applicable to other hybrid systems. It's applicability is in principle as wide as that of the IMM algorithm.

VI. CONCLUSIONS

An attractive MM estimator, called MGS algorithm, has been presented in Part III [10]. It is the first MM estimator with variable structure that is general enough to be applicable to a large class of state estimation problem compounded with structural and/or parametric uncertainties or changes and is easily implementable. The potential improvement in cost-effectiveness of the MGS algorithm on the IMM estimator has been illustrated via a simple example of fault detection and identification in [10]. It has been demonstrated in this paper how this algorithm can be designed via an example of tracking a maneuvering target. General guidelines for such a design have also been presented. Simulation results in terms of e.g., rms state and mode estimation errors, mode probabilities, percentages of correct, incorrect and no mode identification, average distance between the true mode and the models used, and computational requirements, have been presented for both a fairly complicated deterministic scenario and a more realistic random scenario in which the acceleration is a semi-Markov process. They demonstrate that the MGS algorithm is substantially more cost-effective than the IMM estimator using all models: Compared with the IMM algorithm for the example considered, the MGS estimator has either a substantial reduction in computation while having the same performance or a significant improvement in performance while having lower or comparable computational requirements. In view of the great dependence of the performance and computation of an MM algorithm on the test scenarios and performance measures, the random test scenario and the new performance measures proposed will also prove to be useful for the evaluation of many other MM algorithms.

REFERENCES

[1] 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.

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

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

[4] 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, 213–287.

[5] Li, X. R.
Multiple-model estimation with variable structure—Part II: Recursive adaptive model-set approach. Submitted.
An early brief version of its two separate parts has appeared respectively in *Proceedings of the 1998 American Control Conference*, Philadelphia, PA, June 1998, 2439–2443; and in *Proceedings of the 1998 SPIE Conference on Signal and Data Processing of Small Targets*, **3373-58**, Orlando, FL, Apr. 1998, 546–558.

[6] 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.

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

[8] Li, X. R., and He, C. (1999)
Model-set choice and design for multiple-model estimation.
Submitted.

[9] Li, X. R., and Zhang, Y. M. (1999)
Multiple-model estimation with variable structure—Part V: Likely-model-set algorithm. Submitted.
An early brief version appeared in *Proceedings of the 1998 SPIE Conference on Signal and Data Processing of Small Targets*, **3373-26**, Orlando, FL, Apr. 1998, 234–246.

[10] 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). This issue (225–241).

[11] 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.

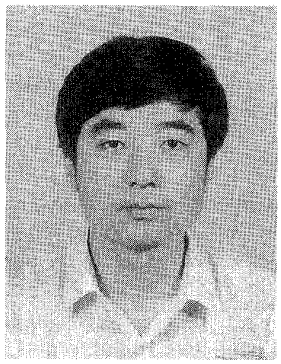
[12] Munir, A., and Atherton, D. P. (1995)
Adaptive interacting multiple model algorithm for tracking a maneuvering target.
IEE Proceedings—Radar, Sonar, and Navigation, **142** (Feb. 1995), 11–17.

[13] Zhang, Y. M., and Li, X. R. (1998)
Detection and diagnosis of sensor and actuator failures using interacting multiple-model estimator.
IEEE Transactions on Aerospace and Electronic Systems, **34** (Oct. 1998), 1293–1313.

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

Dr. Li joined the Department of Electrical Engineering, University of New Orleans, LA, in August 1994. During 1986–1987 he did research in electric power engineering at the University of Calgary, Alberta, Canada. He was an Assistant Professor in the Department of Electrical Engineering, University of Hartford, West Hartford, Connecticut, from 1992 to 1994. His current research interests include signal and data processing, target tracking and information fusion, stochastic systems theory and its applications, and electric power.

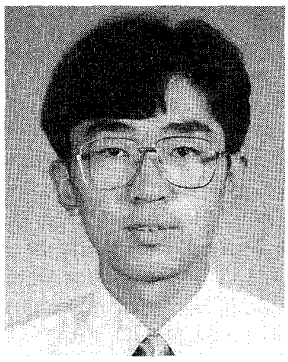
He has published more than 20 refereed journal articles, more than 50 conference papers, three invited book chapters, coauthored a monograph *Multitarget-Multisensor Tracking* (YBS Publishing, Storrs, CT, 1995) and a graduate text *Estimation and Tracking: Principles, Techniques, and Software* (Artech House, Boston, MA, 1993). He serves as an Editor of the IEEE *Transactions on Aerospace and Electronic Systems*. He received the CAREER award and the RIA award from National Science Foundation, and the 1996 Faculty Early Career Achievement Award for Excellence in Research from University of New Orleans. He is the Steering Committee Chair of the First International Conference on Information Fusion. He has been invited to give more than 30 seminars at research institutions in U.S. and other countries. He has won several Outstanding Paper awards.



Youmin Zhang was born in Dali, Shaanxi Province, P.R. China in April 1963. He 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 has been an Associate Professor since 1992 and an Associate Chairman of the Department since 1994. From Sept. 1995 to Oct. 1998, he held several research positions in the Dept. of Electrical Engineering, University of New Orleans, Dept. of Electrical and Computer Engineering, Louisiana State University, and the Dept. of Electrical Engineering, State University of New York at Binghamton, respectively. He is currently with the Dept. of Electrical and Computer Engineering, The University of Western Ontario as a Postdoctoral Research Fellow. His main research interests include fault detection, diagnosis and fault-tolerant control, filtering, estimation and identification and applications to target tracking, aircraft flight data processing, neural networks and applications to modeling, identification and control, robust filtering, identification and control.

Dr. Youmin Zhang was the recipient of several research achievement awards from the National Education Commission of China, Aeronautics and Astronauts Industry Ministry of China; an outstanding teaching achievement award from Shaanxi Province of China; several outstanding teaching awards from Northwestern Polytechnical University; an award of outstanding young teacher from Northwestern Polytechnical University. He served as the General Chairman of the 10th Youth Conference on Automatic Control and Automation in China in 1994 and has served as the international program committee member of several international conferences. He has published more than 20 journal papers and edited one book. He has also received several Outstanding Paper awards.



Xiaorong Zhi was born in Gansu, China on August 7, 1972. He received the B.S. degree in automatic control from Northwestern Polytechnical University, Xi'an, China, in 1993 and the M.S. degree in electrical engineering from University of New Orleans, New Orleans, in 1997.

He is currently an engineer at Ultratech Stepper Inc., San Jose, CA. His research interests include estimation, target tracking, signal processing, and real time control system.