[8] Ludwig, G., and El-Hamamsy, S. (1991)
Coupled inductance and reluctance models of magnetic component.
IEEE Transactions on Power Electronics, **6**, 2 (Apr. 1991), 240–250.

[9] Cuk, S., and Zhang, Z. (1986)
Coupled-inductor analysis and design.
In *IEEE Power Electronics Specialists Conference Record*, 1986, 655–665.

[10] Hamill, D. (1993)
Lumped equivalent circuits of magnetic components: The gyrator-capacitor approach.
IEEE Transactions on Power Electronics, **8**, 2 (Apr. 1993), 97–103.

[11] Chan, J. H., Vladimirescu, A., Gao, X. C., Liebmann, P., and Valainis, J. (1991)
Nonlinear transformer model for circuit simulation.
IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, **10**, 4 (Apr. 1991), 476–482.

[12] Erickson, R. W., and Maksimovic, D. (1998)
A multiple-winding magnetics model having directly measurable parameters.
In *IEEE Power Electronics Specialists Conference Record*, May 1998, 1472–1478.

[13] *Using Matlab Version 5*
The Math Works, Inc., Massachusetts, 1997.

[14] *MN8CX 1–2 MHz Power Ferrite.*
Ceramic Magnetics, Inc., data sheets.

[15] *Caddock Film Resistors Catalog* (26th ed.) (1997)
Caddock Electronics, Inc., 1997.

[16] *HP 4194A Impedance/Gain-Phase Analyzer Operation Manual.*
Hewlett-Packard, April 1986, Tokyo, Japan 192.

[17] Graphical modeling in SABER.
Analogy Inc., 1996, Engineering note 121.

[18] Srinivas, S. (1999)
Frequency-domain modeling of multi-winding magnetics based on the extended cantilever model.
Masters Thesis, University of Florida, Gainesville, 1999.

Numerically Robust Implementation of Multiple-Model Algorithms

Standard implementation of multiple-model (MM) estimation algorithms may suffer from numerical problems, especially numerical underflows, which occur when the true model is vastly different from one or more models used in the algorithm. This may be devastating to the performance of the MM algorithm. Numerical robust implementations of some of the most popular MM algorithms are presented. Simulation results are provided to verify the proposed implementation and to compare with the implementations with a lower bound.

I. INTRODUCTION

Multiple-model (MM) method is a powerful adaptive approach. It has received a great deal of attention in recent years due to its unique power in handling problems with both structural and parametric uncertainties and/or changes and in decomposing a complex problem into simpler subproblems. Its great recent success ranges from target tracking to fault detection and isolation, and from biomedical signal processing to process control (see, e.g., [6]). In his Bode Lecture given at the 1995 IEEE Conference on Decision and Control, Kumpati S. Narendra advocated this powerful, robust, and adaptive method.

In the MM approach, a set of models is designed to represent the possible system behavior patterns or structures, called system modes, and the overall estimate is obtained by a certain combination of the estimates from the filters running in parallel based on the individual models that match (or represent) particular system modes.

The MM method was initiated in [8]. Many applications of this MM estimator can be found in the literature under various names, such as the "multiple model adaptive estimator," "parallel processing algorithm," "partitioned filter," "self-tuning estimator," "filter bank," and "Gaussian sum adaptive filter" [4] (see [6] for the long list of references therein). These names suggest the structure, features, and capability of this "first-generation" MM estimator, to be referred to

Manuscript received November 16, 1998; revised July 3, 1999; released for publication October 6, 1999.

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

IEEE Log No. T-AES/36/1/02603.

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.

as the *static MM (SMM) algorithm.* This early work did not consider jumps in system mode and led to the noninteracting algorithms in which individual model-based filters do not interact with each other. In the more recently developed interacting (or switching, dynamic) algorithms, such as the Generalized Pseudo Bayesian (GPB) and the Interacting Multiple-Model (IMM) estimators [1–3], the system mode is more generally assumed to be able to jump among members of a mode set, which is usually modeled as Markovian switching among models. For details, the reader is referred to the books [1, 2, 16] and the survey work [6, 10].

The standard (i.e., straightforward) implementation of the MM algorithms may have numerical problems in some situations. The most commonly encountered one is that it may lead to numerical underflow or overflow in the calculation of the model probabilities for certain model sets under some scenarios. Such underflows may cause unacceptable performance of the MM algorithms if the true mode undergoes a jump and has a large distance from at least one of the models assumed. A typical application area of MM estimation in which such situations usually occur is fault detection and identification. Such situations may also occur in target tracking. It is also not acceptable for the development of commercial software.

A common technique to circumvent the numerical problems of the standard implementation is to place a lower bound on the model probabilities so as to prevent any model probability from becoming too small. Although this technique enables each model to be practically activated quickly, it is only an ad hoc trick without solid theoretical justification and thus there is no guarantee that it will work well in most cases. For example, how large should the lower bound be? In addition, whenever it is used, the implementation is not an exact implementation of the MM algorithm.

This paper presents numerically robust and exact implementations of the SMM, IMM and GPB1 (i.e., first-order GPB) algorithms. The key idea is to calculate recursively the various model probabilities only in their exponential forms to avoid underflows and overflows. In other words, only the exponents of the model probabilities are calculated. These implementations overcome the numerical problems with the standard implementation. They are also essential for the development of commercial software of MM algorithms.

The remaining of the paper is organized as follows. Section II describes the numerical problems with the standard implementation of the MM algorithms. The numerically robust implementations are derived and presented in Section III. Simulation results of a simple fault detection and identification example are given in Section IV to verify the numerical robust implementation and to compare with other implementations. The paper is concluded with a brief summary of the results.

II. NUMERICAL PROBLEMS WITH STANDARD IMPLEMENTATION

There are several MM algorithms that are popular for adaptive estimation. Among them the Interacting Multiple-Model (IMM) algorithm [3] and the SMM algorithm, sometime known as Multiple-Model Adaptive Estimator (MMAE) (see, e.g., [9]), are the most popular ones.

Consider the following jump linear systems:

$$x_{k+1} = F_k(s_{k+1})x_k + G_k(s_{k+1})u_k + \Gamma_k(s_{k+1})w_k(s_{k+1}) \tag{1}$$

$$z_k = H_k(s_k)x_k + v_k(s_k) \tag{2}$$

where s_k is the true mode (behavior pattern) of the system at time k. Denote by superscript i the quantity pertinent to model m_i. For example,

$$F_k^i = F_k(s_{k+1})|_{s_{k+1}=m_i}.$$

Tables I and II give one cycle of the SMM and IMM algorithms with a model set M for the above simplest stochastic hybrid system, where $z^k \triangleq \{z_1, z_2, \ldots, z_k\}$.

It can be observed that the IMM algorithm is a generalization of the SMM algorithm in the following sense:

$$\text{SMM} = \text{IMM}|_{\mu_{k|k-1}=\mu_{k-1}, \hat{x}^{0i}_{k-1|k-1}=\hat{x}^i_{k-1|k-1}, P^{0i}_{k-1|k-1}=P^i_{k-1|k-1}}. \tag{3}$$

The major numerical problem with the standard implementation of the SMM or IMM algorithm is the possible underflow in the calculation of the model probabilities. This problem usually exists when the true system mode is very different from one or more models used in the algorithm. A typical application area in which such problems arise is fault detection and identification where some models are necessarily quite different from the system mode in effect.

In the SMM algorithm, this problem is caused by the following calculation of the model probability

$$\mu_k^i = \frac{1}{c}L_k^i\mu_{k-1}^i = \frac{1}{c}|2\pi S_k^i|^{-1/2} \times \exp[-\tfrac{1}{2}(z_k - \hat{z}^i_{k|k-1})'(S_k^i)^{-1}(z_k - \hat{z}^i_{k|k-1})]\mu_{k-1}^i \tag{4}$$

where c is a normalization factor. When the true mode is very different from model i, $z_k - \hat{z}^i_{k|k-1}$ is likely to be very large but S_k^i will not be as large as it should be since it is calculated under the assumption that model i is the correct model. As a result, the normalized measurement residual squared $(z_k - \hat{z}^i_{k|k-1})' \cdot (S_k^i)^{-1}(z_k - \hat{z}^i_{k|k-1})$ will be very large and thus

TABLE I
One Cycle of SMM Algorithm with Kalman Filter

1.	Model-conditional filtering $[\forall m_j \in M]$:	
	predicted state:	$\bar{x}^j \triangleq E[x_k \mid m_k^j, z^{k-1}] = F_{k-1}^j \hat{x}_{k-1\mid k-1} + G_{k-1}^j u_{k-1} + \Gamma_{k-1}^j \bar{w}_{k-1}^j$
	predicted covariance:	$\bar{P}^j \triangleq E[(x_k - \bar{x}^j)(x_k - \bar{x}^j)' \mid m_k^j, z^{k-1}]$ $= F_{k-1}^j P_{k-1\mid k-1} (F_{k-1}^j)' + \Gamma_{k-1}^j Q_{k-1}^j (\Gamma_{k-1}^j)'$
	measurement residual:	$\tilde{z}^j \triangleq z_k - E[z_k \mid m_k^j, z^{k-1}] = z_k - H_k^j \bar{x}^j - \bar{v}_k^j$
	residual covariance:	$S^j \triangleq \mathrm{cov}[\tilde{z}^j \mid m_k^j, z^{k-1}] = H_k^j \bar{P}^j (H_k^j)' + R_k^j$
	filter gain:	$K^j = \bar{P}^j (H_k^j)' (S^j)^{-1}$
	updated state:	$\hat{x}_{k\mid k}^j \triangleq E[x_k \mid m_k^j, z^k] = \bar{x}^j + K^j \tilde{z}^j$
	updated covariance:	$P_{k\mid k}^j \triangleq E[(x_k - \hat{x}_{k\mid k}^j)(x_k - \hat{x}_{k\mid k}^j)' \mid m_k^j, z^k] = \bar{P}^j - K^j S^j (K^j)'$
2.	Mode probability update $[\forall m_j \in M]$:	
	likelihood function:	$L^j \triangleq p[\tilde{z} \mid m_k^j, z^{k-1}] = \mathcal{N}[\tilde{z}^j; 0, S^j]$
	mode probability:	$\mu_k^j \triangleq P\{m_k^j \mid z^k\} = \dfrac{\mu_{k-1}^j L^j}{\sum_i \mu_{k-1}^i L^i}$
3.	Combination:	
	overall estimate:	$\hat{x}_{k\mid k} \triangleq E[x_k \mid z^k] = \sum_j \hat{x}_{k\mid k}^j \mu_k^j$
	overall covariance:	$P_{k\mid k} \triangleq E[(x_k - \hat{x}_{k\mid k})(x_k - \hat{x}_{k\mid k})' \mid z^k]$ $= \sum_j [P_{k\mid k}^j + (\hat{x}_{k\mid k} - \hat{x}_{k\mid k}^j)(\hat{x}_{k\mid k} - \hat{x}_{k\mid k}^j)'] \mu_k^j$

TABLE II
One Cycle of IMM Algorithm with Kalman Filter

1.	Model-conditional initialization and reinitialization (mixing) $[\forall m_j \in M]$:	
	predicted mode probability:	$\mu_{k\mid k-1}^j \triangleq P\{m_k^j \mid z^{k-1}\} = \sum_i \pi_{ij} \mu_{k-1}^i$
	mixing weight:	$\mu^{i\mid j} \triangleq P\{m_{k-1}^i \mid m_k^j, z^{k-1}\} = \pi_{ij} \mu_{k-1}^i / \mu_{k\mid k-1}^j$
	mixing estimate:	$\hat{x}^{0j} \triangleq E[x_{k-1} \mid m_k^j, z^{k-1}] = \sum_i \hat{x}_{k-1\mid k-1}^i \mu^{i\mid j}$
	mixing covariance:	$P^{0j} \triangleq E[(x_k - \hat{x}^{0j})(x_k - \hat{x}^{0j})' \mid m_k^j, z^{k-1}]$ $= \sum_i [P_{k-1\mid k-1}^i + (\hat{x}^{0j} - \hat{x}_{k-1\mid k-1}^i)(\hat{x}^{0j} - \hat{x}_{k-1\mid k-1}^i)'] \mu^{i\mid j}$
2.	Model-conditional filtering $[\forall m_j \in M]$:	
	predicted state:	$\bar{x}^j \triangleq E[x_k \mid m_k^j, z^{k-1}] = F_{k-1}^j \hat{x}^{0j} + G_{k-1}^j u_{k-1} + \Gamma_{k-1}^j \bar{w}_{k-1}^j$
	predicted covariance:	$\bar{P}^j \triangleq E[(x_k - \bar{x}^j)(x_k - \bar{x}^j)' \mid m_k^j, z^{k-1}]$ $= F_{k-1}^j P^{0j} (F_{k-1}^j)' + \Gamma_{k-1}^j Q_{k-1}^j (\Gamma_{k-1}^j)'$
	measurement residual:	$\tilde{z}^j \triangleq z_k - E[z_k \mid m_k^j, z^{k-1}] = z_k - H_k^j \bar{x}^j - \bar{v}_k^j$
	residual covariance:	$S^j \triangleq \mathrm{cov}[\tilde{z}^j \mid m_k^j, z^{k-1}] = H_k^j \bar{P}^j (H_k^j)' + R_k^j$
	filter gain:	$K^j = \bar{P}^j (H_k^j)' (S^j)^{-1}$
	updated state:	$\hat{x}_{k\mid k}^j \triangleq E[x_k \mid m_k^j, z^k] = \bar{x}^j + K^j \tilde{z}^j$
	updated covariance:	$P_{k\mid k}^j \triangleq E[(x_k - \hat{x}_{k\mid k}^j)(x_k - \hat{x}_{k\mid k}^j)' \mid m_k^j, z^k] = \bar{P}^j - K^j S^j (K^j)'$
3.	Mode probability update $[\forall m_j \in M]$:	
	likelihood function:	$L^j \triangleq p[\tilde{z} \mid m_k^j, z^{k-1}] = \mathcal{N}[\tilde{z}^j; 0, S^j]$
	mode probability:	$\mu_k^j \triangleq P\{m_k^j \mid z^k\} = \dfrac{\mu_{k\mid k-1}^j L^j}{\sum_i \mu_{k\mid k-1}^i L^i}$
4.	Combination:	
	overall estimate:	$\hat{x}_{k\mid k} \triangleq E[x_k \mid z^k] = \sum_j \hat{x}_{k\mid k}^j \mu_k^j$
	overall covariance:	$P_{k\mid k} \triangleq E[(x_k - \hat{x}_{k\mid k})(x_k - \hat{x}_{k\mid k})' \mid z^k]$ $= \sum_j [P_{k\mid k}^j + (\hat{x}_{k\mid k} - \hat{x}_{k\mid k}^j)(\hat{x}_{k\mid k} - \hat{x}_{k\mid k}^j)'] \mu_k^j$

the likelihood L_k^i will be extremely small, which may lead to underflow by itself or by the cumulative effect of the recursion (4).

In the IMM algorithm, the problem is less severe but underflow may still occur if certain transition probabilities are zero. The model probabilities are updated in the IMM algorithm using the recursion

$$\mu_k^j = \frac{1}{c} L_k^j \mu_{k|k-1}^j = \frac{1}{c} |2\pi S_k^j|^{-1/2} \times \exp[-\tfrac{1}{2}(z_k - \hat{z}_{k|k-1}^j)'(S_k^j)^{-1}(z_k - \hat{z}_{k|k-1}^j)]\mu_{k|k-1}^j$$

where the predicted model probabilities are given by

$$\mu_{k|k-1}^j = \sum_i \pi_{ij} \mu_{k-1}^i.$$

If $\pi_{ij} \triangleq P\{m_k^i \mid m_{k-1}^j\} \neq 0$ and μ_{k-1}^i was not small, then $\mu_{k|k-1}^j$ is not extremely small even if $\mu_{k-1}^j = 0$ and there will be no cumulative effect resulting in underflow and thus an underflow may only be caused by an extremely small (underflow or close to underflow) likelihood value. If, however, $\pi_{ij} = 0$ for all models i whose previous probability μ_{k-1}^i was not small, then $\mu_{k|k-1}^j$ is zero or extremely small and the cumulative effect may lead to an underflow for both $\mu_{k|k-1}^j$ and μ_k^j. When an underflow occurs to $\mu_{k|k-1}^j$, an overflow will occur when calculating the mixing weight $\mu_{k-1|k-1}^{i|j}$.

This paper presents an implementation of the SMM and IMM that avoids the above-mentioned numerical problems.

III. NUMERICALLY ROBUST IMPLEMENTATION

Common practice to avoid the above-mentioned numerical problems of the SMM algorithm is to place a lower bound (say 10^{-4}) on each model probability at each time so that none of the models are "dead" [9, 12, 15]. This trick kills two birds with one stone. It also makes it much faster to revive a model that would be "dead" without such a lower bound (i.e., to increase the model probability to a significant level) if the situation deems it appropriate. This trick is also applicable to the IMM algorithm.

The use of a lower bound reduces the maximum possible probability of a model from 1 to $1 - (N - 1)b$, where N is the number of models used and b is the lower bound value. This will degrade the (steady-state) performance of the MM algorithm at least to some degree if the dominating model actually matches the true mode, which is usually the case most of the time. This is the price paid to have a fast revival capability. Probably more importantly, the performance of the MM algorithm usually depends on how large the lower bound is set to and the lower bound value is usually obtained via a trial-and-error design process. How this value should be set to is affected by many factors, such as how many models are used, how quickly the revival process should be and how good the steady-state performance is expected. There are no general guidelines for choosing the lower bound values.

The IMM algorithm has a built-in fast revival capability due to the fundamental assumption that the true mode of the system may jump. This possibility of mode jumps is accounted for by the use of a non-zero probability of mode transition. Thus even if $\mu_{k-1}^j = 0$, $\mu_{k|k-1}^j$ will still be non-zero provided that the transition probability $\pi_{ij} \neq 0$ for one of the models with a non-zero probability at $k-1$. As a result, it is not necessary to place a lower bound on the model probabilities. Rather, it is more reasonable to place a lower bound (say 10^{-50}) on the transition probabilities to avoid the numerical underflow mentioned above.

The basic idea of our numerically robust implementation of the SMM and IMM algorithms is to put the various model probabilities in their exponential forms and calculate only the exponents. Since SMM algorithm can be obtained from the IMM algorithm using (3), only the numerical robust implementation of the IMM algorithm is derived below.

Assume it is known from cycle $k-1$ that $\mu_{k-1}^i = e^{-a_{k-1}^i}$ or $a_{k-1}^i = -\ln \mu_{k-1}^i$, $\forall i$, are given. Let

$$A_j = a_{k-1}^n = \min\{a_{k-1}^i : \pi_{ij} \neq 0\}. \tag{5}$$

Then,

$$\mu_{k|k-1}^{A_j} \triangleq e^{A_j} \mu_{k|k-1}^j = e^{A_j} \sum_i \pi_{ij} \mu_{k-1}^i = \sum_i \pi_{ij} e^{-(a_{k-1}^i - A_j)} \geq \pi_{nj} > 0. \tag{6}$$

Hence there cannot be an underflow for $\mu_{k|k-1}^{A_j}$ and thus let

$$\mu_{k|k-1}^{A_j} = e^{-\alpha_j} \quad \Rightarrow \quad \alpha_j = -\ln \mu_{k|k-1}^{A_j} > 0. \tag{7}$$

The mixing weights are given by

$$\mu_{k-1}^{i|j} = \frac{\pi_{ij}\mu_{k-1}^i}{\mu_{k|k-1}^j} = \frac{\pi_{ij} e^{A_j} \mu_{k-1}^i}{e^{A_j} \mu_{k|k-1}^j} = \frac{\pi_{ij} e^{-(a_{k-1}^i - A_j)}}{\mu_{k|k-1}^{A_j}} = \pi_{ij} e^{-(a_{k-1}^i - A_j + \alpha_j)} \tag{8}$$

and thus $\mu_{k-1}^{i|j}$ will not have an overflow, although it may have an underflow. The underflow of $\mu_{k-1}^{i|j}$ does not matter at all since $\mu_{k-1}^{i|j}$ is never used in the calculation of the model probabilities and when it underflows the corresponding weight in the estimate and covariance mixing is extremely small and can be treated as zero.

Next let

$$L_k^i = e^{-\beta_i} \tag{9}$$

where L_k^i is the likelihood of model i at k. Then the model probability at k is updated as follows:

$$\mu_k^j = \frac{\mu_{k|k-1}^j L_k^j}{\sum_i \mu_{k|k-1}^i L_k^i} = \frac{e^{A_j}\mu_{k|k-1}^j L_k^j}{e^{A_j}\sum_i \mu_{k|k-1}^i L_k^i}$$

$$= \frac{\mu_{k|k-1}^{A_j} e^{-\beta_j}}{\sum_i e^{A_j} e^{-A_i} \mu_{k|k-1}^{A_i} e^{-\beta_i}} = \frac{e^{-\alpha_j} e^{-\beta_j}}{\sum_i e^{A_j} e^{-A_i} e^{-\alpha_i} e^{-\beta_i}}$$

$$= \frac{1}{\sum_i \exp[(A_j + \beta_j + \alpha_j) - (A_i + \beta_i + \alpha_i)]}.$$

Now let

$$m = \arg\min_j (A_j + \beta_j + \alpha_j). \tag{10}$$

Then it is clear that

$$\mu_k^m = \max_j \mu_k^j \neq 0$$

and thus μ_k^m can be calculated by

$$\mu_k^m = \frac{1}{\sum_i \exp[(A_m + \beta_m + \alpha_m) - (A_i + \beta_i + \alpha_i)]} \tag{11}$$

$$a_k^m \triangleq -\ln \mu_k^m \tag{12}$$

although some (but not all) terms in the denominator of (11) may underflow, which, however, poses no problem here.

Note now that

$$\frac{\mu_k^i}{\mu_k^m} = \frac{\mu_{k|k-1}^i L_k^i}{\mu_{k|k-1}^m L_k^m} = \frac{e^{A_m} e^{A_i} \mu_{k|k-1}^i e^{-\beta_i}}{e^{A_i} e^{A_m} \mu_{k|k-1}^m e^{-\beta m}}$$

$$= \frac{\mu_{k|k-1}^{A_i}}{\mu_{k|k-1}^{A_m}} e^{(A_m+\beta_m)-(A_i+\beta i)} = e^{(A_m+\alpha_m+\beta_m)-(A_i+\alpha_i+\beta i)}.$$

Thus,

$$e^{-a_k^i} \triangleq \mu_k^i = \mu_k^m e^{(A_m+\alpha_m+\beta_m)-(A_i+\alpha_i+\beta i)}$$

which leads to

$$a_k^i = a_k^m + (A_i + \alpha_i + \beta i) - (A_m + \alpha_m + \beta_m), \quad \forall \ i. \tag{13}$$

This completes the cycle.

For the SMM algorithm, the following can be derived similarly:

$$\mu_k^j = \frac{\mu_{k-1}^j L_k^j}{\sum_i \mu_{k-1}^i L_k^i} = \frac{e^{-a_{k-1}^j} e^{-\beta_j}}{\sum_i e^{-a_{k-1}^i} e^{-\beta_i}}$$

$$= \frac{1}{\sum_i \exp[(a_{k-1}^j + \beta_j) - (a_{k-1}^i + \beta_i)]}$$

$$m = \arg\min_j (a_{k-1}^j + \beta_j)$$

$$\mu_k^m = \max_j \mu_k^j = \frac{1}{\sum_i \exp[(a_{k-1}^m + \beta_m) - (a_{k-1}^i + \beta_i)]} \neq 0$$

$$a_k^m = -\ln \mu_k^m$$

$$e^{-a_k^i} = \mu_k^i = \mu_k^m \frac{\mu_{k-1}^i L_k^i}{\mu_{k-1}^m L_k^m} = e^{-a_k^m} \frac{e^{-a_{k-1}^i} e^{-\beta_i}}{e^{-a_{k-1}^m} e^{-\beta_m}}$$

$$= e^{(a_{k-1}^m + \beta_m) - (a_k^m + a_{k-1}^i + \beta_i)}$$

$$a_k^i = a_k^m + (a_{k-1}^i + \beta_i) - (a_{k-1}^m + \beta_m).$$

Note that model m in the last equation above has the highest probability at k but not necessarily at $k-1$; that is, a_{k-1}^m is the negative exponent of the probability of model m at $k-1$ rather than those of the highest model probability at $k-1$. That is, a_k^m and a_{k-1}^m have the same m.

Based on the above derivation, the numerical implementation of the SMM algorithm is simply to replace the mode probability update step with the following

$$\beta_i = \tfrac{1}{2}(\tilde{z}^i)'(S^i)^{-1}\tilde{z}^i + \tfrac{1}{2}\ln|2\pi S^i|$$

$$m = \arg\min_i (a_{k-1}^i + \beta_i)$$

$$\mu_k^m = \frac{1}{\sum_i \exp[(a_{k-1}^m + \beta_m) - (a_{k-1}^i + \beta_i)]}$$

$$a_k^m = -\ln \mu_k^m$$

$$a_k^i = a_k^m + (a_{k-1}^i + \beta_i) - (a_{k-1}^m + \beta_m)$$

$$\mu_k^i \triangleq P\{m_k^j \mid z^k\} = e^{-a_k^i}.$$

The numerically robust implementation of the IMM algorithm is given in Table III. Obviously, dropping the mixing step in the above implementation of the IMM algorithm results in a numerically robust implementation of the GPB1 algorithm.

IV. EXAMPLES OF FAULT DETECTION AND DIAGNOSIS

A. Simple Example

A simple example of fault detection and diagnosis is given first.

Consider the following simple scalar system with dynamics

$$x_{k+1} = 1.1x_k + w_k \tag{14}$$

and two-dimensional measurements

$$z_k = \begin{bmatrix} 0.5 \\ 1.2 \end{bmatrix} x_k + v_k \tag{15}$$

with $w_k \sim \mathcal{N}(0, 0.1^2)$, $v_k \sim \mathcal{N}(0, I)$, and initial condition $x_0 = 0.1$, 1, or 10. The initial state covariance was as $P_0 = 99$ for all scenarios. The initial

TABLE III
One Cycle of the Numerically Robust IMM Algorithm with Kalman Filter

1. Model-conditional initialization and reinitialization (mixing) $[\forall m_j \in M]$:

$$A_j = a_{k-1}^n = \min\{a_{k-1}^i : \pi_{ij} \neq 0\}$$
$$\alpha_j = -\ln\left[\sum_i \pi_{ij} e^{-(a_{k-1}^i - A_j)}\right]$$
$$\mu_{k-1}^{i|j} = \pi_{ij} e^{-(a_{k-1}^i - A_j + \alpha_j)}$$

mixing estimate: $\hat{x}^{0j} \triangleq E[x_{k-1} \mid m_k^j, z^{k-1}] = \sum_i \hat{x}_{k-1|k-1}^i \mu^{i|j}$

mixing covariance: $P^{0j} \triangleq E[(x_k - \hat{x}^{0j})(x_k - \hat{x}^{0j})' \mid m_k^j, z^{k-1}]$
$= \sum_i [P_{k-1|k-1}^i + (\hat{x}^{0j} - \hat{x}_{k-1|k-1}^i)(\hat{x}^{0j} - \hat{x}_{k-1|k-1}^i)']\mu^{i|j}$

2. Model-conditional filtering $[\forall m_j \in M]$:

predicted state: $\bar{x}^j \triangleq E[x_k \mid m_k^j, z^{k-1}] = F_{k-1}^j \hat{x}^{0j} + G_{k-1}^j u_{k-1} + \Gamma_{k-1}^j \bar{w}_{k-1}^j$

predicted covariance: $\bar{P}^j \triangleq E[(x_k - \bar{x}^j)(x_k - \bar{x}^j)' \mid m_k^j, z^{k-1}]$
$= F_{k-1}^j P^{0j} (F_{k-1}^j)' + \Gamma_{k-1}^j Q_{k-1}^j (\Gamma_{k-1}^j)'$

measurement residual: $\tilde{z}^j \triangleq z_k - E[z_k \mid m_k^j, z^{k-1}] = z_k - H_k^j \bar{x}^j - \bar{v}_k^j$

residual covariance: $S^j \triangleq \text{cov}[\tilde{z}^j \mid m_k^j, z^{k-1}] = H_k^j \bar{P}^j (H_k^j)' + R_k^j$

filter gain: $K^j = \bar{P}^j (H_k^j)' (S^j)^{-1}$

updated state: $\hat{x}_{k|k}^j \triangleq E[x_k \mid m_k^j, z^k] = \bar{x}^j + K^j \tilde{z}^j$

updated covariance: $P_{k|k}^j \triangleq E[(x_k - \hat{x}_{k|k}^j)(x_k - \hat{x}_{k|k}^j)' \mid m_k^j, z^k] = \bar{P}^j - K^j S^j (K^j)'$

3. Mode probability update $[\forall m_j \in M]$:

$$\beta_i = \tfrac{1}{2}(\tilde{z}^i)'(S^i)^{-1}\tilde{z}^i + \tfrac{1}{2}\ln|2\pi S^i|$$
$$m = \arg\min_i (A_i + \beta_i + \alpha_i)$$
$$\mu_k^m = \frac{1}{\sum_i \exp[(A_m + \beta_m + \alpha_m) - (A_i + \beta_i + \alpha_i)]}$$
$$a_k^m = -\ln \mu_k^m$$
$$a_k^i = A_i + \alpha_i + \beta i + a_k^m - (A_m + \alpha_m + \beta_m)$$
$$\mu_k^i \triangleq P\{m_k^j \mid z^k\} = e^{-a_k^i}$$

4. Combination:

overall estimate: $\hat{x}_{k|k} \triangleq E[x_k \mid z^k] = \sum_j \hat{x}_{k|k}^j \mu_k^j$

overall covariance: $P_{k|k} \triangleq E[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})' \mid z^k]$
$= \sum_j [P_{k|k}^j + (\hat{x}_{k|k} - \hat{x}_{k|k}^j)(\hat{x}_{k|k} - \hat{x}_{k|k}^j)']\mu_k^j$

state estimate for every model was set to be equal and as a random variable: $\hat{x}_{0|0} \sim \mathcal{N}(x_0, P_0)$. System and measurement noise covariance matrices were set as $Q = 0.1^2$ and $R = \text{diag}(1,1)$. The initial mode probability for each model was set uniformly as $\mu_0^i = 1/5$ for $i = 1,\ldots,5$. The mode transition probability matrix was designed as

$$\Pi = \begin{bmatrix} 0.95 & 0.025 & 0.025 & 0 & 0 \\ 0.05 & 0.90 & 0 & 0.05 & 0 \\ 0.05 & 0 & 0.90 & 0 & 0.05 \\ 0 & 0.1 & 0 & 0.90 & 0 \\ 0 & 0 & 0.1 & 0 & 0.90 \end{bmatrix} \quad (16)$$

Note that the measurements are noisy scaled versions of the state from two separate sensors. Suppose that 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.

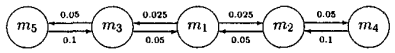


Fig. 1. Topology of total model set.

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

$$m_1 : \begin{bmatrix} 0.5 \\ 1.2 \end{bmatrix} \quad m_2 : \begin{bmatrix} 0.25 \\ 1.2 \end{bmatrix} \quad m_3 : \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix}$$
$$m_4 : \begin{bmatrix} 0 \\ 1.2 \end{bmatrix} \quad m_5 : \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} \quad (17)$$

The topology of the model set assumed is shown in Fig. 1, where the self-transition probabilities are not shown.

Consider the following simple fault scenarios.

S1. The system has no fault throughout the time.

S2. 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 $k = 120$.

TABLE IV
Comparison of Various Implementation of SMM and IMM Algorithms for Scenario S1

	CID (%)	IID (%)	NID (%)	RMS Error	Max Error	Flops	Modal Distance	Mode Error
$x_0 = 0.1$:								
SMM	80.94	7.74	11.32	0.4266	1.078	984	0.0658	0.0747
SMM-nr	79.89	14.68	5.43	0.4074	0.6263	1084	0.1384	0.1464
SMM-lb	80.81	7.34	11.84	42.900	620.6	1089	0.0699	0.0779
IMM	77.36	14.11	8.53	0.4812	1.078	1247	0.0825	0.0972
IMM-nr	81.56	3.99	14.45	0.4427	1.222	1336	0.1011	0.1095
IMM-lb	83.14	1.99	14.87	0.4267	1.116	1394	0.0724	0.0829
$x_0 = 1$:								
SMM	93.46	3.41	3.13	0.4031	1.138	984	0.0222	0.0240
SMM-nr	93.42	2.65	3.96	0.4059	1.043	1084	0.0297	0.0296
SMM-lb	93.42	3.36	3.21	186.9	2715	1089	0.0248	0.0261
IMM	92.22	4.82	2.96	0.4314	1.138	1244	0.0264	0.0337
IMM-nr	92.98	2.63	4.39	0.4060	1.229	1325	0.0321	0.0334
IMM-lb	94.54	1.66	3.79	0.4014	1.175	1394	0.0225	0.0251
$x_0 = 10$:								
SMM	99.92	0.08	0	0.3783	1.259	984	0.000397	0.001002
SMM-nr	99.92	0.08	0	0.3783	1.259	1084	0.000397	0.001002
SMM-lb	99.92	0.08	0	1817	2.643e4	1089	0.002932	0.002902
IMM	99.92	0.08	0	0.3786	1.259	1241	0.000396	0.001007
IMM-nr	99.92	0.08	0	0.3785	1.248	1312	0.000433	0.001079
IMM-lb	99.92	0.08	0	0.3784	1.236	1394	0.000407	0.001052

TABLE V
Comparison of Various Implementation of SMM and IMM Algorithms for Scenario S2

	CID (%)	IID (%)	NID (%)	RMS Error	Max Error	Flops	Modal Distance	Mode Error
$x_0 = 0.1$:								
SMM	Diverge							
SMM-nr	71.77	22.81	5.43	548.2	8.447e3	1084	0.1946	0.2027
SMM-lb	80.19	7.97	11.84	640.4	1.348e4	1089	0.0738	0.0819
IMM	Diverge							
IMM-nr	81.56	3.99	14.45	80.55	2.202e3	1336	0.1011	0.1095
IMM-lb	83.14	1.99	14.87	0.47	1.116	1394	0.0724	0.0829
$x_0 = 1$:								
SMM	Diverge							
SMM-nr	85.30	10.78	3.93	2397	3.696e4	1084	0.0859	0.0859
SMM-lb	92.80	3.99	3.21	2801	5.897e4	1089	0.0288	0.0301
IMM	Diverge							
IMM-nr	92.98	2.63	4.39	351.3	9.633e3	1325	0.0321	0.0334
IMM-lb	94.54	1.66	3.79	0.4445	1.175	1394	0.0225	0.0251
$x_0 = 10$:								
SMM	Diverge							
SMM-nr	91.80	8.20	0	2.333e4	3.597e5	1084	0.0567	0.0573
SMM-lb	99.30	0.70	0	2.726e4	5.739e5	1089	0.0069	0.0069
IMM	Diverge							
IMM-nr	99.92	0.08	0	3417	9.376e4	1312	0.0004331	0.001079
IMM-lb	99.92	0.08	0	0.4214	1.236	1394	0.0004073	0.001052

S3. The system has no fault from $k = 0$ to $k = 160$, except that a sensor 1 or 2 (with equal probability) total failure occurs at a random time $k = \tau_1$ and lasts for a random period of time τ_2, where $\tau_1 \sim \mathcal{N}(100, 5^2)$ and $\tau_2 \sim \mathcal{N}(20, 2^2)$.

The fault detection and identification results over the time period considered depend to a large degree on the initial state of the system because the noise levels relative to the magnitude of the state vary with the initial value. Tables IV–VI give comparison results

TABLE VI
Comparisons of Various Implementation of SMM and IMM Algorithms for Scenario S3

	CID (%)	IID (%)	NID (%)	RMS Error	Max Error	Flops	Modal Distance	Mode Error
$x_0 = 0.1$:								
SMM	Diverge							
SMM-nr	73.01	21.59	5.40	738.3	4.718e3	1084	0.1752	0.2175
SMM-lb	81.75	7.37	10.88	935.8	6.867e3	1089	0.0695	0.0824
IMM	Diverge							
IMM-nr	82.71	4.17	13.12	115.1	1.011e3	1335	0.0995	0.1034
IMM-lb	84.21	1.81	13.99	0.4484	1.093	1394	0.0705	0.0765
$x_0 = 1$:								
SMM	Diverge							
SMM-nr	85.28	11.15	3.57	3190	2.317e4	1084	0.0740	0.1173
SMM-lb	93.03	4.18	2.79	4036	3.218e4	1089	0.0293	0.0422
IMM	Diverge							
IMM-nr	93.05	3.00	3.95	497.2	4.399e3	1325	0.0339	0.0360
IMM-lb	94.23	2.08	3.69	0.4347	1.089	1394	0.0245	0.0280
$x_0 = 10$:								
SMM	Diverge							
SMM-nr	91.76	8.24	0	3.145e4	2.232e5	1084	0.0423	0.0847
SMM-lb	99.54	0.46	0	3.989e4	3.206e5	1089	0.0051	0.0151
IMM	Diverge							
IMM-nr	99.87	0.13	0	4.937e3	4.334e4	1312	0.0005017	0.001159
IMM-lb	99.88	0.12	0	0.4037	0.9496	1394	0.0004673	0.001117

of the SMM and IMM algorithms with standard implementation (SMM and IMM), numerically robust implementation (SMM-nr and IMM-nr) and with lower bound (SMM-lb and IMM-lb), for the initial state $x_0 = 0.1$, $x_0 = 1$, and $x_0 = 10$, respectively, where SMM-lb had a lower bound 10^{-3} on the mode probabilities and IMM-lb had a lower bound 10^{-300} on the mode transition probabilities.

In the tables, 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); 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; it is indecisive (NID) if no model has a probability above the threshold; the average modal distance and mode error over L Monte Carlo runs are defined, respectively,

$$\text{average modal distance at time } k = \frac{1}{L}\sum_{n=1}^{L}\sum_{m_i}\|s_k - m_i\|P\{s_k = m_i \mid z^k\} \tag{18}$$

$$\text{rms mode error at } k = \sqrt{\frac{1}{L}\sum_{n=1}^{L}\|s_k - \hat{s}_{k|k}\|^2} \tag{19}$$

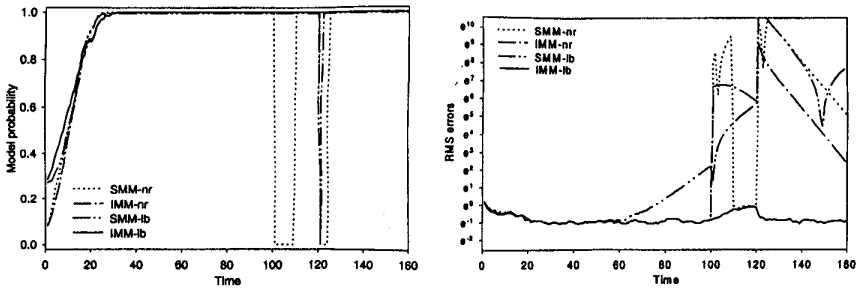
where n is the Monte Carlo run index, $\|s_k - m_i\|$ is the Euclidean distance between the true mode and the model m_i, and the mode estimate is defined by

$$\hat{s}_{k|k} = \sum_{m_i} m_i P\{s_k = m_i \mid z^k\}.$$

The reader is referred to [7] for a more complete discussion of these performance measures.

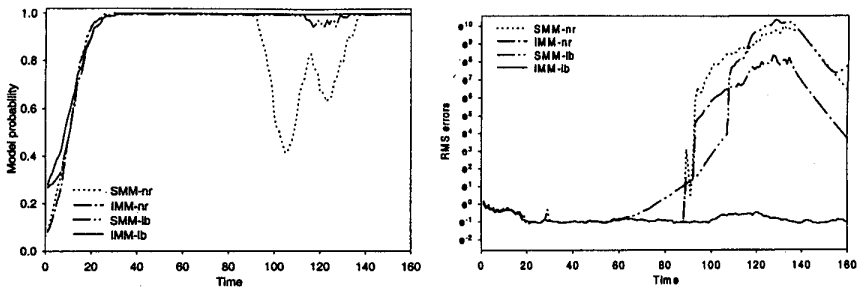
Figs. 2 and 3 show, for $x_0 = 1$, the probabilities of the correct model (i.e., the model that corresponds to the true mode in effect at the given time) and the rms errors of the two implementations of the SMM and IMM algorithms that did not diverge under scenarios S2 and S3, respectively, over 100 Monte Carlo runs. Under scenario S1, all three implementations of the SMM and IMM algorithms had probability curves similar to that of the IMM-lb in Fig. 2(a). Their rms errors are similar to that of the IMM-lb in Fig. 2(b) except that SMM-lb was diverging after time 60.

Scenario S1 illustrates that the use of a lower bound on mode probabilities in the SMM algorithm may lead to an unacceptable estimation error of the base state. It is clear that under all the scenarios, 1) for the SMM algorithm, the implementation with a lower bound has the best detection and identification results but worst base state estimates; 2) for the IMM algorithm, the implementation with a lower bound has the best results. Under scenarios S2 and S3 in which the system mode undergoes sudden jumps, the standard implementations of the SMM and IMM algorithms both diverge.



(a) (b)

Fig. 2. Correct model probabilities and rms errors of various implementations of SMM and IMM algorithms under scenario S2. (a) Probabilities of correct model. (b) rms errors.



(a) (b)

Fig. 3. Correct model probabilities and rms errors of various implementations of SMM and IMM algorithms under scenario S3. (a) Probabilities of correct model. (b) rms errors.

B. Two Aircraft Examples with Sensor and Actuator Failures

1) *Aircraft Model*: Two types of aircraft models were used for the sensor and actuator fault detection and diagnosis (FDD) in [17] (two other papers that use IMM algorithm for FDD are [5, 11]). These examples are used here to evaluate the performance of the different implementations of the MM algorithms.

The dynamics of both an F/A-18 aircraft [14] in a given region of the flight envelope, level flight at 10,000 ft with a speed of Mach 0.6, and of another high performance aircraft [13] with a high subsonic cruise speed (Mach = 0.8) at an altitude of 35,000 ft, can be linearized and their motion can be described by the continuous-time state variable equations

$$\dot{x}(t) = Ax(t) + Bu(t) + \xi_c(t) \tag{20}$$

$$z(t) = Cx(t) + \eta(t). \tag{21}$$

For F/A-18 aircraft, the model has eight (four longitudinal and four lateral) state variables with the longitudinal and lateral motions completely decoupled. It is represented by state vector $x = [u\ w\ q\ \theta\ v\ r\ p\ \varphi]'$, where u,v,w represent velocities in forward, lateral, and vertical directions of the body axes, respectively; p,q,r represent roll, pitch, and yaw angular rates, respectively; θ, φ are pitch and roll angles, respectively. The aircraft utilizes five pairs of control surfaces (some of which can be used symmetrically and asymmetrically) to achieve seven different control inputs, represented by $U = [\delta_e\ \delta_{sle}\ \delta_{ste}\ \delta_{ast}\ \delta_{ate}\ \delta_a\ \delta_r]'$. The seven control input variables, three for longitudinal control and four for lateral control, are symmetric stabilator (or elevator, denoted by δ_e), symmetric leading edge flap (δ_{sle}), symmetric trailing edge flap (δ_{ste}), asymmetric stabilator (δ_{ast}), asymmetric trailing edge flap (δ_{ate}), aileron (δ_a) and rudder (δ_r). The system and control matrices at the given normal flight condition are represented, respectively, by A and B given in [17]. A more detailed description of the F/A-18 aircraft can be found in [14].

For the second aircraft, the model has eight (four longitudinal and four lateral) state variables, with the longitudinal and lateral motions completely decoupled, represented by state vector $x = [\alpha\ q\ u\ \theta\ \beta\ p\ r\ \varphi]'$, where u represents forward velocity; p,q,r represent roll, pitch, and yaw angular rates, respectively; α,β denote angle of attack and angle of sideslip, respectively; θ, φ are pitch and roll angles, respectively. The control input vector is represented by $U = [\delta_{EL}\ \delta_{ER}\ \delta_{CL}\ \delta_{CR}\ \delta_{SL}\ \delta_{SR}\ \delta_A\ \delta_R]'$. They are left and right elevators (denoted by δ_{EL} and δ_{ER}), left and right canards (δ_{CL} and δ_{CR}), left and right spoilers (δ_{SL} and δ_{SR}), aileron (δ_A), and rudder (δ_R). Only two control

TABLE VII
Performance Comparison of FDD for Simultaneous Sensor and Actuator Failures

	Algorithm	CID	FA	IID	MFD	NID	Delay	Flops
Case 1	IMM	94.50	0.005	0.025	0.065	5.41	0.413	9.121
	IMM-nr	96.52	0.02	0.075	0.165	3.22	0.490	8.822
	IMM-lb	94.50	0.005	0.025	0.065	5.41	0.413	9.121
	SMM	Diverge						
	SMM-nr	61.62	13.35	0	24.5	0.53	0	4.767
	SMM-lb	20.61	34.22	11.08	0.71	33.38	4.387	4.836
Case 2	IMM	94.34	0.005	0.095	0.035	5.53	0.439	9.121
	IMM-nr	96.59	0.015	0.115	0.08	3.19	0.469	8.822
	IMM-lb	94.34	0.005	0.095	0.035	5.53	0.439	9.121
	SMM	Diverge						
	SMM-nr	61.62	13.35	0	24.5	0.53	0	4.767
	SMM-lb	20.62	34.34	11.22	0.64	33.18	4.492	4.836
Case 3	IMM	92.30	0	0	1.02	6.68	1.228	9.121
	IMM-nr	90.84	0.005	0.02	4.41	4.725	0.855	8.828
	IMM-lb	92.30	0	0	1.02	6.68	1.228	9.121
	SMM	Diverge						
	SMM-nr	61.60	13.29	0	24.5	0.61	0	4.767
	SMM-lb	16.27	33.97	8.38	1.46	39.92	3.630	4.836

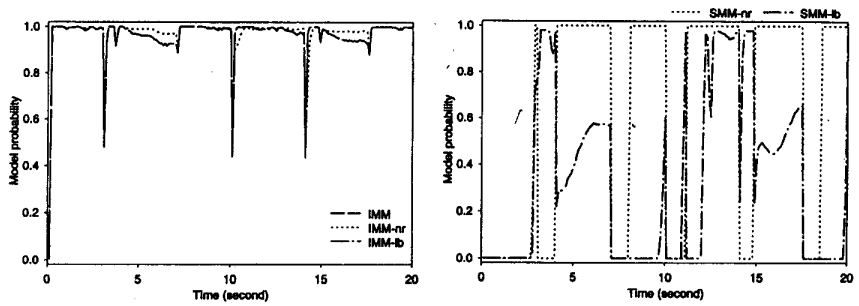


Fig. 4. Correct model probabilities and various implementations of SMM and IMM algorithms. (a) Implementations of IMM algorithms. (b) Implementations of SMM algorithms.

inputs, left elevator δ_{EL} and right elevator δ_{ER}, are relevant to the longitudinal movement of the aircraft under the normal flight condition. The system and control matrix A, B are given in [17]. A more detailed description of the aircraft can be found in [13]. It was assumed for simplicity that all the state components are directly measurable and C is an identity matrix.

2) *FDD with Deterministic Test Scenario for Second Aircraft Example*: The FDD of longitudinal aircraft sensor and actuator failures under a deterministic test scenario is considered here. To simulate different and complicated failures situations, the total and partial sensor/actuator failures, simultaneous partial sensor and actuator failures were included in one deterministic test scenario. The designed fault sequence is: total pitch rate failure between $k = 31$ and $k = 40$; simultaneous 20% partial angle of attack and right elevator failures between $k = 71$ and 80; a 40% partial left elevator failure between $k = 101$ and $k = 111$; a 40% partial pitch angle failure between $k = 141$ and $k = 149$; and a total left elevator failure between $k = 176$ and $k = 185$. The designed set of models which includes the normal and possible fault modes consists of 21 models: one normal, four total sensor failure, four 40% partial sensor failure, two total actuator failure, two 40% partial actuator failure, and eight simultaneous sensor and actuator failure models. The initial state estimate for every model was set to be equal and as a random variable: $\hat{x}_{0|0} \sim \mathcal{N}(x_0, 99I)$, where $x_0 = [5, 10, 50, 5]'$. $Q = 0.01^2 I_{2\times 2}$ and $R = 0.2^2 I_{3\times 3}$. The initial mode probability for each model was set to be equal. The mode transition probability matrix was designed based on the mean sojourn time $\tau_{\text{normal}} = 3$ s and $\tau_{\text{fault}} = 1$ s for all test sceanrios: $\pi_{11} = 29/30$, $\pi_{1i} = 1/600$, $i \neq 1$ and $\pi_{jj} = 9/10$, $\pi_{j1} = 1/10$ and zero for the other elements. The reader is referred to [17] for more details of the design parameters used.

The FDD performance indices and the flops (in 10^4) are given in Table VII. The corresponding correct model probabilities for case 1 are plotted in Fig. 4. It should be noted that, in addition to the performance

TABLE VIII
Performance Comparison for Random Fault Scenarios

Fault	Design	Algorithm	CID	FA	IID	MFD	NID	Flops
actuator	1	IMM-nr	96.10	1.01	1.66	1.23	0	0.7789
		IMM-lb	95.48	1.13	2.19	1.20	0	0.7792
		SMM-nr	66.12	10.78	2.65	20.45	0	0.6925
		SMM-lb	87.86	5.97	4.52	1.66	0	0.6916
	2	IMM-nr	94.72	1.52	2.98	0.78	0	1.392
		IMM-lb	94.40	1.74	3.21	0.65	0	1.401
		SMM-nr	27.66	47.88	20.84	3.62	0	1.157
		SMM-lb	80.66	9.34	8.53	1.465	0.005	1.158
sensor	1	IMM-nr	92.46	3.11	1.58	2.175	0.675	2.845
		IMM-1b	86.02	8.245	3.43	1.725	0.585	2.888
		SMM-nr	66.46	20.43	8.175	4.005	0.925	2.095
		SMM-1b	66.20	20.72	8.26	3.985	0.825	2.103
	2	IMM-nr	87.68	6.455	3.395	1.765	0.705	6.652
		IMM-1b	84.81	8.975	4.145	1.355	0.715	6.843
		SMM-nr	45.84	34.46	15.07	2.905	1.73	3.996
		SMM-lb	45.70	34.80	15.16	2.88	1.45	4.039

indices, CID, IID, and NID, given in the above simple example, more FDD performance indices, such as the false alarm (FA) and missed fault detection (MFD), have been adopted from [17] and used in these two aircraft examples. Case 1 represents the situation in which the model set and the noise statistics are known exactly. Case 2 differs from Case 1 in that a 5% modeling error exists in the system dynamics. In Case 3, the noise matrices Q and R used for the filter are 4 times of the true ones so that the performance impact due to the uncertainties in noise statistics can be evaluated. It is clear that the numerically robust implementations are significantly better than the other implementations and the SMM algorithms have much worse results than the IMM algorithm. However, the numerically robust implementation of the SMM algorithm, SMM-nr, has much better performance than the one with a lower bound, SMM-lb. In Table VII, "delay" stands for the latency from the time the true mode changes to the time it is detected and correctly identified among the cases in which the system mode is (eventually) identified. It is interesting to note that the standard implementation of the SMM algorithm in this example gave extreme performance only; it either completely failed to detect or identify the mode change correctly or it provided a detection and correct identification without any delay.

3) *FDD with Random Test Scenarios for F/A-18 Aircraft Example*: In this example, a number of random test scenarios have been designed for the evaluation of the proposed numerically robust implementations. Actuator or sensor failures with total and partial faults were designed separately. The magnitude of a partial fault was designed as uniformly distributed over $(0,1)$. Since the FDD performance depends largely on model-set design, two choices of the quantization of the partial fault magnitude as $M_1 = \{0, 0.5, 1\}$ and $M_2 = \{0, 0.25, 0.5, 0.75, 1\}$ were considered. Other quantization level may of course be used. The FDD performance indices and the flops ($\times 10^4$) are given in Table VIII for these two designs. Design 1 used quantization M_1 while Design 2 used M_2. For example, for the case of a sensor fault, Design 1 used 9 models (one normal, four 50% partial sensor failure, four total sensor failures) and Design 2 used 17 models (one normal, four 25% partial sensor failure, four 50% partial sensor failure, four 75% partial sensor failure, four total sensor failures). For an actuator fault, only primary control surface failure was considered. This leads to 3 models for Design 1 and 5 models for Design 2. Essentially the same design parameters as those in Subsection IVB2 were used except $x_0 = [500, 100, 10, 5]'$, $Q = 0.01^2 I_{3\times 3}$, $R = 0.2^2 I_{4\times 4}$, and that the transition probability matrix is different for different model sets. The reader is referred to [17, 18] for more details of the design parameters used.

It is obvious from Table VIII that the performance indices, especially CID, vary with the design of the model set.

It should be noted that the performance indices given above should not be applied to different model set designs; otherwise they may be misleading: the more models, the worse the performance. From fault detection and isolation point of view, the above performance indices are for each of the possible fault and normal modes. It may be more interesting in some practical applications to know the performance indices for each faulty actuator or sensor, rather than for each mode. Thus, the above CID, FA, IID, MFD, and NID are also calculated based on the fault models corresponding to each sensor (or actuator) instead of based on each fault mode. As such, average modal distance was calculated only over the modes for the

TABLE IX
Performance Comparison With Modified Indices

Fault	Design	Algorithm	CID	FA	IID	MFD	NID	$\mathcal{D}$
actuator	1	IMM-nr	97.80	0	0.97	1.23	0	0.0337
		IMM-lb	97.30	0	1.50	1.20	0	0.0333
		SMM-nr	76.90	0	2.65	20.45	0	0.0645
		SMM-lb	95.59	0	2.74	1.67	0	0.0647
	2	IMM-nr	97.10	0	2.12	0.78	0	0.0209
		IMM-lb	97.00	0	2.345	0.65	0.005	0.0214
		SMM-nr	78.62	0	17.76	3.62	0	0.1305
		SMM-lb	91.38	0	7.15	1.465	0.005	0.0532
sensor	1	IMM-nr	89.52	0	1.58	8.23	0.675	0.0341
		IMM-lb	88.92	0	3.195	7.305	0.585	0.0907
		SMM-nr	83.32	0	6.845	8.905	0.925	0.1323
		SMM-lb	83.38	0	6.93	8.86	0.825	0.1323
	2	IMM-nr	90.81	0	3.25	5.23	0.705	0.0594
		IMM-lb	90.59	0	4.00	4.695	0.715	0.0637
		SMM-nr	78.82	0	14.31	5.14	1.73	0.0901
		SMM-lb	79.07	0	14.40	5.08	1.45	0.1323

corresponding sensor (or actuator), given by

$$\mathcal{D} = \frac{1}{L}\sum_{n=1}^{L}\left\{\frac{1}{k_{\max}}\sum_{k=1}^{k_{\max}}\left\{\sum_{m_i\in M}\|s_k - m_i\|P\{s_k = m_i \mid s_k \in M, z^k\}\right\}\right\}$$

where M represents the model set consisting of the partial and total fault models pertaining to a given sensor (or actuator); and L is the number of Monte Carlo runs.

The corresponding FDD results are given in Table IX. It can be seen that similar results were obtained for the two model-set designs. Compared with the results in Table VIII, much better performance was demonstrated for SMM-nr and SMM-lb. But the performance of the IMM algorithm is still much better than that of the SMM. For both sets of performance indices, the numerically robust implementation of the IMM algorithm, IMM-nr, has better performance than the one with a lower bound on model transition probabilities, IMM-lb.

The reader is referred to [17] for robustness analysis and performance comparison of the different MM-based FDD approaches to the design of model transition probabilities, modeling errors, and noise statistics.

V. CONCLUSIONS

It has been shown that numerical problems, especially numerical underflows, may devastate the performance of MM estimation algorithms with a standard (i.e., straightforward) implementation if the true mode of the system undergoes a jump and has a large distance from some of the models assumed. Numerically robust implementations of the SMM, IMM, and GPB1 algorithms have been proposed in this paper. The key to their robustness is that various model probabilities are calculated recursively only in their exponential forms. That is, only their exponents are calculated recursively. Simulation results of several fault detection and identification examples illustrate the following. The "steady-state" performance of the numerically robust implementation of the SMM algorithm is significantly superior to the implementation with a lower bound on the model probabilities; whereas the latter may outperform the former if the true mode undergoes frequent jumps. For the IMM algorithm, the numerically robust implementation and the one with a lower bound on mode transition probabilities are substantially superior to the standard implementation. No general conclusion can be drawn for the IMM algorithm as to whether the numerically robust implementation is better than the implementation with a lower bound—it is problem dependent.

X. RONG LI
Department of Electrical Engineering
University of New Orleans
New Orleans, LA 70148
E-mail: (xli@uno.edu)

YOUMIN ZHANG
Department of Electrical and Computer Engineering
The University of Western Ontario
London, Ontario, N6A 5B9
Canada
E-mail: (ymzhang@julian.uwo.ca)

REFERENCES

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

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

[3] 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.

[4] 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.

[5] Efe, M., and Atherton, D. P. (1997)
The IMM approach to the fault detection problem.
Presented at the 11th IFAC Symposium on System Identification, Fukuoka, Japan, July 1997.

[6] 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.

[7] Li, X. R., and He, C.
Model-set design and choice for multiple-model estimation.
To be published.

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

[9] Maybeck, P. S., and Hanlon, P. D. (1995)
Performance enhancement of a multiple model adaptive estimator.
IEEE Transactions on Aerospace and Electronic Control, **31** (Oct. 1995), 1240–1254.

[10] Mazor, E., Averbuch, A., Bar-Shalom, Y., and Dayan, J. (1996)
Interacting multiple model methods in target tracking: A survey.
IEEE Transactions on Aerospace and Electronic Control, **34**, 1 (1996), 103–123.

[11] Mehra, R. K., Rago, C., and Seereeram, S. (1997)
Failure detection and identification using a nonlinear interactive multiple model (IMM) filtering approach with aerospace applications.
In *Proceedings of 11th IFAC Symposium on System Identification*, Fukuoka, Japan, July 1997.

[12] Menke, T. E., and Maybeck, P. S. (1995)
Sensor/actuator failure detection in the Vista F-16 by multiple model adaptive estimation.
IEEE Transactions on Aerospace and Electronic Control, **31** (Oct. 1995), 1218–1229.

[13] Napolitano, M., and Swaim, R. (1992)
New techniques for aircraft flight control reconfiguration.
In C. Leondes, Ed., *Control and Dynamic Systems*.
New York: Academic Press, 1992, 155–228.

[14] Rauch, H., and Kline-Schoder, R. (1993)
Fault detection, isolation, and reconfiguration for aircraft using neural networks.
In *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, Aug. 1993, 1527–1537.

[15] Watanabe, K. (1989)
A multiple model adaptive filtering approach to fault diagnosis in stochastic systems.
In R. J. Patton, P. M. Frank, and R. N. Clark, Eds., *Fault Diagnosis in Dynamic Systems: Theory and Applications*.
Englewood Cliffs, NJ: Prentice-Hall, 1989, 411–438.

[16] Watanabe, K. (1992)
Adaptive Estimation and Control: Paritioning Approach.
New York: Prentice-Hall, 1992.

[17] Zhang, Y. M., and Li, X. R. (1998)
Detection and diagnosis of sensor and actuator failures using IMM estimator.
IEEE Transactions on Aerospace and Electronic Control, **34** (Oct. 1998), 1293–1312.

Ranging Airport Pseudolite for Local Area Augmentation

This paper discusses the integration of an airport pseudolite (APL) into a local area augmented differential GPS (DGPS) based precision approach system. A prototype architecture is described that is being used to develop requirements for the Local Area Augmentation System (LAAS). Key features of this prototype system are presented along with its current performance. Key features discussed include the use of a multipath limiting antenna (MLA), APL signal structure factors, a unique APL automatic gain control (AGC), and GPS blanking technique to maximize APL tracking performance, while minimizing the electromagnetic interference (EMI) to nominal DGPS performance.

BACKGROUND

The concept of ground-based GPS-like transmissions, now called pseudolites, has been around ever since the early development of the GPS (i.e., 1977 Yuma, AZ test) [1]. Research over the past seven years has concentrated on the addition of an airport pseudolite (APL) to increase the availability of a Local Area Augmentation System (LAAS). It has been demonstrated that a code-based system, with code and carrier measurements being passed on a VHF data link [2], can provide the accuracy, continuity, and integrity for a CAT III system. In order to realize an increase in availability with the addition of an APL(s) to a LAAS, the APL must be of high quality and able to be used as a ranging source. Availability is a qualitative indication of the ability of the navigation systems to provide the required level of performance at the beginning of the intended precision approach operation. This level of performance for

Manuscript received November 10, 1998; revised September 17, 1999; released for publication October 6, 1999.

Refereeing of this contribution was handled by P. K. Willett.

IEEE Log No. T-AES/36/1/02604.