Generalized SSPRT for Fault Identification and Estimation of Linear Dynamic Systems Based on Multiple Model Algorithm

Ji Zhang[1], Yu Liu*[2] and Xuguang Li[3]

[1] Department of Computer, North China Electric Power University
Baoding, Hebei 071003, China
[2] Department of Electrical Engineering, University of New Orleans,
New Orleans, LA 70148, USA
lyu2@uno.edu
[3]Clinical Laboratory, 323 Hospital
Xi'an, Shaanxi 710054, China

ABSTRACT
The generalized Shiryayev sequential probability ratio test (SSPRT) is applied to linear dynamic systems for single fault isolation and estimation. The algorithm turns out to be the multiple model (MM) algorithm considering all the possible model trajectories. In real application, this algorithm must be approximated due to its increasing computation complexity and the unknown parameters of the fault severeness. The Gaussian mixture reduction is employed to address the problem of computation complexity. The unknown parameters are estimated in real time by model augmentation based on maximum likelihood estimation (MLE) or expectation. Hence, the system state estimation, fault identification and estimation can be fulfilled simultaneously by a multiple model algorithm incorporating these two techniques. The performance of the proposed algorithm is demonstrated by Monte Carlo simulation. Although our algorithm is developed under the assumption of single fault, it can be generalized to deal with the case of (infrequent) sequential multiple faults. The case of simultaneous faults is more complicated and will be considered in future work.

Keywords: Generalized SSPRT, state estimation, fault isolation and estimation, multiple model, Gaussian mixture reduction, model augmentation

1. Introduction

Fault diagnosis has been extensively studied [1-10]. It can be addressed by hardware redundancy or analytical redundancy [4]. With the increasing computational power and decreasing cost of the digital signal processors and software, the method of analytical redundancy, which diagnoses the possible fault by comparing the signals from a real system with a mathematical model, is prevailing due to its low cost and high flexibility.

In this work, we study the fault diagnosis problem of a linear stochastic system subject to a single sensor/actuator fault. As pointed out in [3], the fault diagnosis problem consists of the following three sub-problems:

Fault detection: Is there a fault in the system? The answer is "yes", "no" or "unknown". It is actually a change detection problem with binary hypotheses. We want to detect the fault after its occurrence as quick as possible. For simple hypotheses with independent and identically distributed (i.i.d.) observations, optimal algorithms exist, e.g., cumulative sum (CUSUM) test [11], which (asymptotically) minimizes the worst-case expected detection delay under some false alarm restrictions [12, 13]. If the change time is assumed random and prior information is available, the Shiryayev sequential probability ratio test (SSPRT) [14] in Bayesian framework is optimal in terms of a Bayesian risk. Note that the fault source needs not to be identified (or isolated) in fault detection.

Fault isolation (or identification): Usually there are many components in modern systems and merely knowing there is a fault (by fault detection) in a system is far from enough for a quick and effective remedy. Hence, the goal of fault isolation is to identify the source of a fault as soon as possible. So, the answer to this problem is a fault type, e.g., which component in a system is faulty. This is a change detection problem with multiple alternative hypotheses. Each alternative hypothesis corresponds to a fault model (unlike fault detection, all the fault models are included in a single alterative hypothesis) and we need to isolate which hypothesis happens after a change. The fault isolation can be carried out subsequently after the fault detection (i.e., identifying the source after declaration of a fault) or individually (i.e., operates in a stand-along mode without the fault detection). Sequential algorithms for this problem were proposed in [6, 9] and it minimizes the worst-case expected isolation delay under the restriction of mean time before a false alarm/isolation. The generalized SSPRT (GSSPRT) in Bayesian framework was proposed [8] and it is optimal in terms of a Bayesian risk for fault isolation. To the authors' best knowledge, joint optimal solution for fault detection and isolation is not known.

Fault estimation: Estimate the severeness of a fault. For example, a sensor/actuator may fail completely (it does not work at all) or partially (has degenerated performance). This piece of information is useful for future decision and action. If a sensor/actuator fails completely, the system may have to be stopped until the faulty component is replaced or fixed. In another hand, if only minor partial fault occurs, the sensor/actuator may be kept in use, with some online compensation [3].

In this work, system state estimation, fault isolation and estimation are tackled simultaneously. We start from applying the GSSPRT to a linear dynamic system and the algorithm turns out to be a multiple model (MM) algorithm considering all possible model sequences [15]. Fault diagnosis by multiple model algorithms is gaining attentions. A bank of mathematical models is constructed to model the normal operation mode and the fault modes. Filters based on these models are running in parallel, and the system state estimation is obtained by the outputs from MM, and the fault isolation can be done by comparing the model probabilities with the pre-defined thresholds. So, the fault diagnosis and state estimation can be done simultaneously, and the performance of the state estimation is independent to that of fault diagnosis. The MM is attractive since it uses a bunch of models rather than a single model to represent the faulty behaviours of a system [10]. The autonomous multiple model (AMM) [15] algorithm was the first MM algorithm proposed for fault diagnosis [16–18]. However the underlying assumption of AMM about the model trajectory—the model in effect does not change over time—does not fit the change detection problem. Then, the interacting multiple model (IMM) [19] algorithm, which considers the interaction between models, was proposed [7,10,20,21] and they outperform AMM in general. In [7], the IMM was directly applied to a fault isolation problem, but it was assumed that the fault models are exactly known. The hierarchical IMM [22] and IM3L [10] were proposed to address the fault isolation and estimation. However, they were tackled in a sequential manner, i.e., isolation-then-estimation. Without the information of the fault severeness, the isolation performance suffers. In this paper, we propose a multiple model algorithm that solves the fault isolation and estimation simultaneously.

In practical applications, each sensor/actuator fault can be total or partial. So, a parameter $\alpha \in [0, 1]$ is introduced for each sensor/actuator to indicate its fault severeness [7,10]. $\alpha = 0$ means complete failure while $0 < \alpha < 1$ denotes a partial fault. If α is known after a fault occurrence, the hypothesis for each sensor/actuator fault becomes a simple hypothesis and the GSSPRT algorithm for this case is exactly the same as the MM algorithm considering all possible model sequences. Hence, the optimality of GSSPRT and the virtues of MM algorithms are all preserved. Further, the system state can be also estimated as a by-product. However, two difficulties impede its exact implementation in practice:

a) Its computational complexity is increasing due to the increasing number of model sequences.

b) α is unknown in general.

The first problem can be solved by pruning and merging the model sequences so that the total number of sequences is bounded. A bunch of algorithms were proposed for this purpose, e.g., B-best and GPBn, see [15] and the references therein. We propose to use the Gaussian mixture reduction [23, 24] to merge the "similar" model sequences, since it can be better justified than GPBn or IMM. Second, the unknown parameter α can be estimated online by model augmentation. We introduce one augmented model for each sensor/actuator with an (unknown) fault parameter α , which is updated at every step by maximum likelihood estimation (MLE) or expectation—the so-called maximum-likelihood model augmentation (MMA) [10,20] or expected model augmentation (EMA) [15,25], respectively. So, before declaration of a fault, the fault severeness has been estimated and updated in real time based on these augmented models. Further, these augmented models are expected to be "close" to the truth, and hence further benefit the overall performance of state estimation and fault isolation. As shown in the simulation, the EMA and MMA have their pros and cons to each other. The MMA has better adaptation of parameter change and hence has faster isolation and smaller miss detection rate, while the EMA performs better in terms of the parameter estimation and correct isolation rate.

Although our algorithm is developed based on the assumption of single fault, it can be extended to deal with infrequent sequential multiple faults easily provided the interval between two faults is long enough for isolating the first fault before the second fault. If a fault has been identified and its fault severeness estimated, then all the models can be revised to accommodate this fault and hence detection for further fault can be carried out. The case of simultaneous faults is more complicated and deserves further studies.

This paper is organized as follows. First, the problem of fault isolation and estimation for linear dynamic systems is formulated in Sec. 2. The algorithm based on GSSPRT is derived in Sec. 3. The multiple model methods based on the Gaussian mixture reduction and the model augmentation are presented in Sec. 4. Three illustrative examples are provided in Sec. 5, and our algorithms are compared with the IMM method. Conclusions are made in Sec. 6.

2. Problem Formulation

A linear stochastic system subject to a sensor/actuator fault can be formulated as the following first-order Markov jump-linear hybrid system:

$$x_k = F_{k-1}x_{k-1} + B_{k-1}^j u_{k-1} + G_{k-1}w_{k-1} \quad (1)$$

$$z_k = H_k^j x_k + v_k \quad (2)$$

where x_k and z_k are the system state and the measurement at time k, respectively. Each column of B_k is an "actuator" while each row of H_k is a "sensor" in the system [7, 10, 21]. The superscript j denotes that the matrices dependent the model m^j in effect. It is assumed that there are total M fault models $\{m^1, m^2, \cdots, m^M\}$ and one normal model m^0. The control input u_k is assumed deterministic and known all the time. The process noise w_k and measurement noise v_k are Gaussian white noise with zero mean and covariances Q_k and R_k, respectively. The model sequence $\{m_k\}$ is assumed to be a first-order Markov sequence with the transition probability

$$\pi_{ji} \triangleq P\{m_k^i | m_{k-1}^j\}$$

where m_k^i denotes the event that m^i is in effect at time *k*. The system usually starts with the normal model m_0^0 and at each time k it has probability $\pi_{0i} > 0$ to transfer to m^i (the ith fault model). Further, it is assumed that all fault models m^i, $i = 1, \cdots, M$, are absorbing state in the Markov chain, that is

$$\pi_{ji} = \begin{cases} 1, for\ i = j \neq 0 \\ 0, for\ i \neq j, j \neq 0 \end{cases}$$

meaning that once a system gets into one of the fault models, it remains, since we only consider the case with single fault. The possibility that a system recovers automatically from a fault model is ignored since it rarely happens in practice. So, the transition probability matrix (TPM) is

$$\pi = \begin{bmatrix} \pi_{00} & \pi_{01} & \pi_{02} & \cdots & \pi_{0M} \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (3)$$

The total and partial sensor/actuator faults are considered, and the fault models proposed in [7, 10] are adopted. Let $\alpha_k^j \in [0,\ 1)$ denotes the severeness of the jth sensor/actuator's fault at time k. The corresponding "actuator" in B_k or"sensor" in H_k is multiplied by α_k^j due to the fault. Clearly $\alpha_k^j = 0$ means that the sensor/actuator fails completely while $0 < \alpha_k^j < 1$ indicates a partial fault. In general, if a fault happens, α_k^j is unknown and time varying. The problem becomes much more difficult if α_k^j is fast changing. For simplicity, only a constant or a slowly drifting sequence is considered for α_k^j. If prior information is available, more sophisticated dynamic models for α_k^j are also optional.

Under these problem settings, we are trying to achieve the following three goals simultaneously based on sequentially available measurements $Z^k = \{z_1, \mathrm{z}_2, \cdots, z_k\}$:

(a) State estimation ($\hat{x}_{k|k}$) in real time;

(b) Fault identification (i.e., sensor/actuator $\hat{j}$);

(c) Fault severeness estimation $\hat{\alpha}_k^{\hat{j}}$ when a fault is identified.

Once a fault has been isolated and an estimate of the fault severeness is provided, additional actions based on these results can be taken to further inspect the decision and improve the estimation accuracy. This is problem dependent and there are many options, e.g., use a different model set specifically designed for the faulty sensor/actuator to achieve better state estimation and fault estimation. Also, the declared fault can be further tested against the normal model (or other fault models) to mitigate the possible false alarm (or false isolation) rate. We do not further examine these possibilities since they are beyond the scope of this paper.

3. Generalized Shiryayev Sequential Probability Ratio Test

First, we only consider the goal (b) and assume α_k^j is exactly known after a fault occurs. Then an optimal solution in Bayesian framework—generalized Shiryayev sequential probability ratio test (GSSPRT)—was proposed in [8] for this fault isolation problem. The optimality of GSSPRT was proved in terms of a Bayesian risk. Further, it minimizes the time of fault isolation given the costs of false alarm, false isolation and a measurement at each time k, see [8] for the proof and details.

Assume the system starts with no fault. At each time k, the GSSPRT computes the posterior probability of the event θ_k^j □ {Transition from m^0 to m^i occurs at or before time k}, and compares it with a threshold $\mu^{T_i},\ \mathrm{i} = 1, 2, \cdots, \mathrm{M}$. Once one of the thresholds is exceeded, the corresponding fault is declared. The optimal thresholds can be determined by the given costs [8]. Note that θ_k^0 means the system remains in normal mode up to time k. The event θ_k^i is equivalent to the event m_k^i since the fault model m^i is an absorbing state in the Markov chain. The GSSPRT algorithm is summarized as follows:

Declare a sensor/actuator fault i if

$$\mu_k^i \triangleq P\{\theta_k^i | Z^k\} = P\{m_k^i | Z^k\} > \mu^{T_i}$$

$$i = 1, 2, \cdots, M$$

Else, compute $\mu_{k+1}^i, i = 0, 1, \cdots, M$

$$\text{Loop } \mathrm{k} = \mathrm{k} + 1$$

Note that in the algorithm there is no threshold set for m^0 since declaration of the normal model is of no interest. The posterior probability μ_k^i is computed by

$$\mu_k^i = \sum_{h=1}^{n_k^i} \mu_k^{i,(h)}, \qquad \mu_k^{i,(h)} \triangleq P\left\{m_k^{i,(h)} | Z^k\right\}$$

and $m_k^{i,(h)}$denotes a model sequence which starts from time 0 and reaches model m^i at time k , h is the index, and n_k^i is the total number of such sequences. Since the system starts with no fault, $m_k^{0,(h)}$ is the only valid sequence at time k = 0, and hence $\mu_0^{0,(1)} = 1$. The probability $\mu_k^{i,(h)}$ can be computed recursively. Since

$$\left\{m_k^{i,(h)}\right\}_{h=1,\cdots,n_k^i} = \left\{m_k^i, m_{k-1}^{j,(l)}\right\}_{j=0,\cdots,M}^{l=1,\cdots,n_{k-1}^j}$$

by Bayesian formula, we have

$$\mu_k^{i,(h)} = \frac{f(z_k | m_k^{i,(h)}, Z^{k-1})\pi_{ji}\mu_{k-1}^{j,(l)}}{f(z_k | Z^{k-1})}$$

where the likelihood function $f\left(z_k \middle| m_k^{i,(h)}, Z^{k-1}\right)$ can be calculated based on the Kalman filter (KF) [26] under the linear Gaussian assumption. This can be done since the model trajectory has been specified by $m_k^{i,(h)}$,

$$f\left(z_k \middle| m_k^{i,(h)}, Z^{k-1}\right) = N(z_k; H_k^i \bar{x}_{k|k-1}^{(h)}, S_k^{(h)})$$

where

$$\bar{x}_{k|k-1}^{(h)} = E[x_k | m_k^{i,(h)}, Z^{k-1}]$$

$$S_k^{(h)} = H_k^i \hat{P}_{k|k-1}^{(h)} (H_k^i)' + R_k$$

and $\hat{P}_{k|k-1}^{(h)}$ is the error covariance matrix of $\bar{x}_{k|k-1}^{(h)}$. The model conditional density of the system state is

$$f\left(x_k \middle| m_k^i, Z^k\right) = \frac{1}{\mu_k^i} \sum_{h=1}^{n_k^i} f(x_k | m_k^{i,(h)}, Z^k) \mu_k^{i,(h)} \qquad (4)$$

Clearly, $f\left(x_k \middle| m_k^i, Z^k\right)$ is a Gaussian mixture density and the number of Gaussian components is increasing geometrically (i.e., $\left(M+1\right)^k$) with respect to k for a general TPM. However, due to the assumption that all the fault models are absorbing states (the special structure of Eq. (3)), it only increases linearly (i.e., *MK*+1) for our problem.

The estimate of x_k and its mean square error (MSE) matrix can be obtained by:

$$\hat{x}_k \triangleq E[x_k | Z^k] = \sum_{i=0}^{M} \hat{x}_k^i \mu_k^i$$

$$\hat{P}_k = \sum_{i=0}^{M} [\hat{P}_k^i + (\hat{x}_k^i - \hat{x}_k)(\hat{x}_k^i - \hat{x}_k)'] \, \mu_k^i$$

where

$$\hat{x}_k^i \triangleq E[x_k | m_k^i, Z^k]$$

$$\hat{P}_k^i \triangleq \mathrm{MSE}(\hat{x}_k | m_k^i, Z^k)$$

can be obtained from $f\left(x_k \middle| m_k^i, Z^k\right)$ (Eq. (4)). $\hat{x}_k$ is optimal in terms of the MSE.

The above procedure turns out to be the well-known cooperating multiple model (CMM) algorithm [15] considering all possible model trajectories. Before, the multiple model algorithm was developed for state estimation with model uncertainties. It is estimation oriented. Here, it is derived from a totally different angle by starting from GSSPRT for decision purpose. So, the goal (a) and (b) can be fulfilled simultaneously and optimally in terms of their criterions, respectively. Further, the state estimation is not affected by the performance of fault isolation.

Even a false alarm or false isolation occurs, the state estimation is still reliable, since the detection has no impact (or feedback) to the model-conditioned estimates and model probabilities, and hence does not affect the state estimation.

Besides the increasing computational complexity, the unknown fault parameters α_k^j in practical applications incur further difficulties to our algorithm. The hypothesis of a fault model becomes composite in this case, rendering it much more complicated. Further, it is usually very desirable that a good estimate of α_k^j can be provided at the time a fault is identified, meaning that α_k^j should be estimated online along with the fault identification process. The fault estimation can also provide useful information for fault isolation, and consequently benefits its performance.

4. Multiple Model Algorithm Based on Gaussian Mixture Reduction and Model Augmentation

First, we address the problem of computation complexity. As aforementioned, Eq. (4) is a Gaussian mixture density and its number of components is increasing rapidly. Each Gaussian component in $f\left(x_k \middle| m_k^i, Z^k\right)$ corresponds to a model trajectory $m_k^{i,(h)}$. In real time implementation, the number of the components must be bounded.

In MM method, there are many algorithms proposed to reduce the number of model sequences [15]. They can be classified as: a) methods based on hard decision, such as the B-Best algorithm, which keeps the most likely one or a few model sequences and prunes the rest; b) methods based on soft decision, such as the GPBn algorithm, which merges those sequences with common model trajectories in last n steps (they may have different trajectories in older times). In general, the algorithms based on soft decision outperform those based on hard decision. However, for GPBn methods there is no solid ground to justify why sequences with common model steps should be merged. These common parts of model trajectories do not necessarily imply that the corresponding Gaussian components are “close” to each other. Consequently the merging may not lead to good estimation accuracy. We propose to use a more sophisticated scheme based on Gaussian mixture reduction, which involves both pruning and merging. The idea is simple and better justified, but requires more computation. However, in our problem, the number of model trajectories increases linearly instead of geometrically. The reduction process usually needs not to be performed frequently. Of course, this approach can be implemented for general MM algorithm, which may require the reduction for every step.

Once the number of components in $f\left(x_k \middle| m_k^i, Z^k\right)$ exceeds a threshold, the extremely unlikely components can be pruned first. Then, the number of Gaussian components is further reduced to a pre-specified number by pairwise merging, with the grand mean and covariance maintained. This is the so-called Gaussian mixture reduction problem and was studied by [23, 24, 27–32]. The problem is to reduce the number of Gaussian components in a Gaussian mixture density by minimizing the “distance” (to be defined) between the original density and reduced density, subject to the constraint that the grand mean and covariance are unaltered. The optimal solution requires solving a high dimensional constrained nonlinear optimization problem that the weights, means and covariances are chosen such that the “distance” between the original mixture and the reduced mixture is minimized. This is still an open problem and optimal solution is computationally infeasible for most applications. However, a suboptimal and efficient solution is acceptable for our problem. As proposed in [23, 24, 30], a top-down reduction algorithm based on greedy method is employed. Two of the components are selected to merge by minimizing the “distance” between them at each iteration, until the number of components reduces to a pre-determined threshold. For two Gaussian components with weights w_i, means μ_i and covariances P_i, they are merged by

$$w = w_1 + w_2, \quad \mu = \frac{1}{w}(w_1\mu_1 + w_2\mu_2)$$

$$P = \frac{1}{w}\sum_{i=1}^{2} w_i[P_i + (\mu_i - \mu)(\mu_i - \mu)']$$

so that the grand mean and covariance are preserved.

Further, there were many distances proposed for merging. They can be categorized to two classes: global distance and local distance. The global distance of two Gaussian components measures the difference between the original mixture density and the reduced mixture density (by merging these two elements), while the local distance only measures the difference between these two components. The global distance is preferred in general since it considers the overall performance. Kullback-Leibler (KL) divergence may be a good choice [30], but it cannot be evaluated analytically between two Gaussian mixtures, see [33] and reference therein for some numerical methods. An upper bound of KL divergence was proposed in [30] to serve as the distance, which is, however, a local distance. We adopt the distance proposed in [23, 24]—the integral squared difference (ISD). For two Gaussian mixture densities $f(x)$ and $g(x)$, the ISD is defined as

$$D = \int (f(x) - g(x))^2 dx$$

It is a global distance and can be evaluated analytically between any two Gaussian mixture densities:

$$D = J_{ff} + J_{gg} + J_{fg}$$

where

$$J_{ff} = \sum_{i}^{n_f} \sum_{j}^{n_f} w_i w_j \, \mathcal{N}(\mu_i; \mu_j, P_i + P_j)$$

$$J_{gg} = \sum_{i}^{n_g} \sum_{j}^{n_g} \bar{w}_i \bar{w}_j \, \mathcal{N}(\bar{\mu}_i; \bar{\mu}_j, \bar{P}_i + \bar{P}_j)$$

$$J_{fg} = \sum_{i}^{n_f} \sum_{j}^{n_g} w_i \bar{w}_j \, \mathcal{N}(\mu_i; \bar{\mu}_j, P_i + \bar{P}_j)$$

and $\{w_i, \ \mu_i, \ P_i\}$ and $\{\bar{w}_i, \ \bar{\mu}_i, \bar{P}_i\}$ are the weights, means and covariances, respectively, of the i th Gaussian components in $f(x)$ and $g(x)$. Efficient algorithms to compute the distance was proposed in [24]. Hence, the distance between two Gaussian components for merging is defined as

$$D_{ij}^{l} = \int (f(x) - f_{ij}^{l}(x))^2 dx$$

where $f(x)$ is the original Gaussian mixture, $f_{ij}^{l}(x)$ is the mixture density after merging the i th and j th components in $f^{l}(x)$, which is the reduced Gaussian mixture at iteration l. For each iteration l, two components are selected to merge such that D_{ij}^{l} is minimized. The iteration stops when the number of the components reduces to the pre-specified number. Compare with the merging method in GPBn, the merging based on this Gaussian mixture reduction is better justified. Although the above reduction procedure does not reduce the Gaussian component optimally, it is based on a good guidance—only components that are "close" to each other are merged and hence the loss should be smaller than GPBn.

In MM algorithm, at time $k-1$, assume the Gaussian mixture densities $f(x_{k-1}|m_{k-1}^{j}, Z^{k-1})$ and the model probabilities $\mu_{k-1}^{j}$for *j = 0, 1, … , M* are obtained. Then $f(x_k|m_k^j, Z^k)$ can be updated recursively:

$$f(x_k|m_k^j, Z^k) = \frac{f(z_k|m_k^i, x_k) f(x_k|m_k^i, Z^{k-1})}{f(z_k|m_k^i, Z^{k-1})}$$

and the model probability μ_k^i

$$\mu_k^i = \frac{f(z_k|m_k^i, Z^{k-1}) \mu_{k|k-1}^{i}}{f(z_k|Z^{k-1})}$$

where

$$f(x_k|m_k^i, Z^{k-1})$$

$$= \sum_{j=0}^{M} f(x_k|m_k^i, m_{k-1}^j, Z^{k-1}) f(m_{k-1}^j|m_k^i, Z^{k-1}) \qquad (5)$$

$$f(x_k|m_k^i, m_{k-1}^j, Z^{k-1})$$

$$= \int f(x_k|m_k^i, x_{k-1}) f(x_{k-1}|m_{k-1}^j, Z^{k-1}) dx_{k-1} \qquad (6)$$

and

$$\mu_{k|k-1}^{i} \triangleq f(m_k^i|Z^{k-1}) = \sum_{j=0}^{M} \pi_{ji} \mu_{k-1}^{j}$$

$$f(z_k|m_k^i, Z^{k-1}) = \int f(z_k|m_k^i, x_k) f(x_k|m_k^i, Z^{k-1}) dx_k$$

$$f(m_{k-1}^j|m_k^i, Z^{k-1}) = \frac{\pi_{ji}\mu_j}{\sum_j \pi_{ji}\mu_j}$$

$f(x_k|m_k^i, x_{k-1})$ and $f\left(z_k\middle|m_k^i, x_k\right)$ are obtained from Eqs. (1) and (2), respectively. It can be seen from Eq. (5) that the number of the Gaussian components in $f\left(x_k\middle|m_k^i, Z^k\right)$ is increasing in each cycle. The Gaussian mixture reduction is implemented if necessary and $f(x_k|m_k^i, Z^k)$ is replaced by the reduced density.

As mentioned before, due to the special structure of the TPM in our problem, the number of the Gaussian components increases linearly and hence the Gaussian reduction procedure needs not to be performed frequently.

The second problem of applying the MM algorithm is the unknown parameter α_k^j for each sensor/actuator in the fault identification process. Before any fault occurrence, $\alpha_k^j = 1$ and it subjects to a possible sudden jump to a value that

$$0 \le \alpha_k^j < 1 \tag{7}$$

Practically, only when α_k^j drops below a threshold $\alpha^{T,j} < 1$ it is considered as a fault:

$$0 \le \alpha_k^j < \alpha^{T,j}$$

Several methods may be applied to estimate α_k^j. It may be augmented into the system state. However, this requires a dynamic model for α_k^j, and the system becomes nonlinear (for sensor fault models) and subjects to a constraint. It can be also estimated by least squares method with fading memories [34, 35]. However, the abrupt change of α_k^j when a fault happens can incur difficulties for this algorithm. The method based on the maximum likelihood estimation (MLE) [10] is a good choice. For the j th sensor/actuator fault, a few models with different but fixed α_k^j ($\alpha_k^j = \alpha^{j,i}, i = 1,2,\cdots, I,$ where I is the number of fault models for the j th sensor/actuator) values and one augmented model with the parameter α_k^j estimated by the MLE in real time are included in the model set. The estimate $\hat{\alpha}_k^j$ can be obtained by

$$\hat{\alpha}_k^j = \max_{\alpha} f(z_k|\alpha, m_k^j, Z^{k-1})$$

s. t. Eq. (7)

where, for simplicity, $f\left(z_k\middle|\alpha, m_k^j, Z^{k-1}\right)$ can be approximated by a single Gaussian density, e.g., for a sensor fault, it yields

$$f(z_k|\alpha, m_k^j, Z^{k-1}) \approx \mathcal{N}(\tilde{z}_k; 0, S_k)$$

where

$$\tilde{z}_k = z_k - H_k^j(F_{k-1}\hat{x}_{k-1}^j + B_{k-1}^0 u_{k-1})$$

$$S_k = H_k^j \hat{P}(H_k^j)' + R_k$$

$$\hat{P} = F_{k-1}\hat{P}_{k-1}(F_{k-1})' + G_{k-1}Q_{k-1}(G_{k-1})'$$

Since both $\tilde{z}_k$ and S_k are functions of α_k^j, the MLE becomes a one-dimensional nonlinear inequality-constrained optimization problem which may be solved numerically. The MLE for an actuator fault can be obtained similarly, but the optimization procedure is simpler since it becomes a quadratic programming problem, which can be solved analytically by solving a linear equation. Then, the augmented models are updated by α_k^j in real time. This is the maximum likelihood model augmentation (MMA). It has a quick adaption in fault estimation when a fault occurs.

Another option is the expected model augmentation (EMA). It is similar to the MMA but the parameters α_k^j for the augmented models are estimated by weighted average of the models for the same sensor/actuator. That is, for a sensor/actuator fault j, the $\hat{\alpha}_k^j$ for the augmented model is obtained by

$$\hat{\alpha}_k^j = \frac{1}{c}\left(\alpha_k^0 \mu_{k|k-1}^0 + \sum_{i=1}^{I} \alpha^{j,i} \mu_{k|k-1}^{j,i} + \hat{\alpha}_{k-1}^j \mu_{k|k-1}^{\hat{\alpha}_{k-1}^j}\right)$$

where

$$c = \mu_{k|k-1}^0 + \sum_{i=1}^{I} \mu_{k|k-1}^{j,i} + \mu_{k|k-1}^{\hat{\alpha}_{k-1}^j}$$

is a normalizing constant, $\alpha_k^0 \equiv 1$, $\mu_{k|k-1}^{\hat{\alpha}_{k-1}^j}$ is the predicted probability of the augmented model, $\mu_{k|k-1}^{j,i}$ are the predicted model probabilities of the models with fixed parameter $\alpha^{j,i}$. Note that this method circumvents the requirement of the constraint (Eq. (7)) since it is automatically guaranteed by the convex sum.

The augmented models in the model set serve two purposes: a) They are expected (hopefully) to be closer to the truth than the other models, and hence benefit the overall performance of the MM filter. b) The fault severeness can be obtained based on those augmented models, i.e., once a particular fault is declared, the corresponding $\hat{\alpha}_k^j$ can be outputted as the fault estimate.

It is clear that the state estimation, fault isolation and estimation are solved simultaneously the a MM algorithm based on Gaussian mixture reduction and model augmentation. The results provide a basis for further diagnosis and actions. If, for example, a fault is declared but not sever or the faulty sensor/actuator is not critical, some online compensation can be applied to maintain the system in operation. Major action may have to be taken if a crucial or total failure occurs.

5. Illustrative Examples

We provide three illustrative examples to demonstrate the applicability and performance of our algorithms by comparing with the results of IMM based on Monte Carlo (MC) simulation.

5.1 Simulation Scenario

The ground truth is adopted from [10, 21], i.e., a longitudinal vertical take-off and landing (VTOL) of an aircraft. The state is defined as

$$x = [V_h, V_v, q, \theta]$$

where the components are horizontal velocity (m/s), vertical velocity (m/s), pitch rate (rad/s) and pitch angle (rad), respectively. The target dynamic matrix and measurement matrix are obtained by discretization of the continuous system:

$$F_k = e^{AT}$$

$$B_k = \left(\int_0^T e^{AT} d\tau\right) G$$

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Where $T = 0.1s$ is the sampling interval, and

$$A = \begin{bmatrix} -0.0366 & 0.0271 & 0.0188 & -0.4555 \\ 0.0482 & -1.01 & 0.0024 & -4.0208 \\ 0.1002 & 0.3681 & -0.707 & 1.420 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} 0.4422 & 0.1761 \\ 3.5446 & -7.5922 \\ -5.52 & 4.49 \\ 0 & 0 \end{bmatrix}$$

The control input is set to be $u_k = [0.2 \;\; 0.05]'$, the true initial state $= [250 \;\; 50 \;\; 1 \;\; 0.1]'$, the covariances of the process noise $Q_k = 0.2^2 I$ and the measurement noise $R_k = diag[1, 1, 0.1, 0.1]$. In this system, there are two actuators (A1 and A2) and four sensors (S1 – S4). The simulation lasts for 70 steps and the fault occurs at $k = 10$. The initial density of the state for the algorithms is chosen to be a Gaussian density $N(x_0, \hat{P}_0)$ with $\hat{P}_0 = diag[10, 10, 1, 1]$.

5.2 Performance Measures

The performances of our algorithms are evaluated by the following measures:

1.- Correct identification (CI) rate: the rate that an algorithm correctly identifies a fault after it happens.

2.- False identification (FI) rate: the rate that an algorithm incorrectly identifies a fault after it happens.

3.- False alarm (Fa) rate: the rate that an algorithm declares a fault before any fault happens.

4.- Miss detection (MD) rate: the rate that an algorithm fails to declare a fault within the total steps of each MC run.

5.- Average delay (AD): the average delay (in terms of the number of sampling steps) for a correct isolation.

6.- $\breve{\alpha}$: the root mean square error (RMSE) of $\hat{\alpha}$ for a correct isolation.

All the performances are obtained by Monte Carlo simulation with 1000 runs. The thresholds for decision are determined by the results of simulation such that different methods have (almost) the same false alarm rate.

5.3 Total Failure

In this case, we consider the complete failure for each actuator/sensor. Hence, the fault severeness $\alpha = 0$ is known. We compare the IMM, the exact MM (MME) method (i.e., considering all possible model trajectories) and the MM method based on the Gaussian mixture reduction (MMR). All the methods have the same model set of $M+1$ models. It includes one normal model, and one fault model for each sensor/actuator.

There is no model-set mismatch between the algorithms and the ground truth. In MMR, the number of the Gaussian components for each model is reduced to 2 if it exceeds 10. This is a relative simple scenario since the fault parameter is known. The results are given in Table 1. It is clear that a sensor fault is much easier than an actuator fault to be identified.

A sensor fault can be detected (almost) immediately after the occurrence while it takes some time to detect an actuator fault. This makes sense because a sensor fault is directly revealed by the measurement while an actuator fault affects the measurement only through the system state.

The performance differences among the three algorithms for a sensor fault are negligible, while MME and MMR evidently outperform the IMM algorithm for an actuator fault. But this superior performance is achieved at the cost of higher computational demands (given in Table 2). Comparing with MME, the performance loss of MMR is tiny.

Fault	Algo.	CI	Fa	FI	MD	AD
A1	MME	0.854	0.010.	0	0.136	11.01
	MMR	0.850	0.010	0	0.140	11.09
	IMM	0.819	0.010	0.001	0.170	11.94
A2	MME	0.823	0.016	0.057	0.104	15.8
	MMR	0.823	0.016	0.057	0.104	15.8
	IMM	0.771	0.016	0.039	0.174	17.3
S1	MME	0.985	0.015	0	0	0
	MMR	0.985	0.015	0	0	0
	IMM	0.985	0.015	0	0	0
S2	MME	0.990	0.010	0	0	0.36
	MMR	0.990	0.010	0	0	0.36
	IMM	0.990	0.010	0	0	0.39
S3	MME	0.992	0.008	0	0	0
	MMR	0.992	0.008	0	0	0.002
	IMM	0.992	0.008	0	0	0.002
S4	MME	0.990	0.010	0	0	0
	MMR	0.990	0.010	0	0	0
	IMM	0.990	0.010	0	0	0

Table 1. Total fault, $\alpha = 0$ and known.

IMM	MME	MMR
1	20.7	15.4

Table 2. Average computational cost (normalized).

5.4 Random Scenario

In this case, for each MC run, the fault severeness α_k^j is sampled uniformly from the interval [0, $\alpha^{T,j}$] when a fault occurs and remains constant. The IMM, EMA and MMA (both based on Gaussian mixture reduction) are implemented and evaluated. The IMM contains 13 models: one normal model, two fault models for each sensor/actuator with α = 0 and 0.5, respectively. The EMA and MMA contain 19 models: all the models in IMM algorithm and one augmented model for each sensor/actuator. The results are given in Table 3. Similar to Case 1, a sensor fault is easier to be identified than an actuator fault, revealed by a shorter detection delay and better fault estimates. The performance differences among the three algorithms are insignificant for sensor faults. For actuator faults, MMA has a shorter detection delay and smaller miss detection rate, while EMA is better in terms of correct identification rate, false identification rate and estimation root mean square error $\breve{\alpha}$.

This can be explained by the quick adaption of the change in α by MLE after the fault occurrence. It helps identifying the fault faster, but in general its estimation is less accurate than EMA and hence may increase the false identification rate. Overall, they outperform the IMM method.

Fault	Algo.	CI	Fa	FI	MD	AD	$\breve{\alpha}$
A1	MMA	0.709	0.009	0.066	0.216	18.2	0.208
	EMA	0.711	0.009	0.037	0.243	18.4	0.199
	IMM	0.707	0.009	0.045	0.239	18.7	0.209
A2	MMA	0.762	0.011	0.116	0.111	15.1	0.230
	EMA	0.771	0.011	0.099	0.119	17.1	0.215
	IMM	0.743	0.011	0.118	0.128	17.5	0.219
S1	MMA	0.992	0.008	0	0	0	0.103
	EMA	0.992	0.008	0	0	0.257	0.117
	IMM	0.992	0.008	0	0	0.500	0.206
S2	MMA	0.981	0.012	0.007	0	1.320	0.147
	EMA	0.984	0.012	0.004	0	1.322	0.138
	IMM	0.980	0.012	0.008	0	1.320	0.235
S3	MMA	0.990	0.010	0	0	0	0.131
	EMA	0.990	0.010	0	0	0.001	0.126
	IMM	0.990	0.010	0	0	0	0.235
S4	MMA	0.991	0.009	0	0	0	0.135
	EMA	0.991	0.009	0	0	0	0.127
	IMM	0.991	0.009	0	0	0	0.232

Table 3. Random fault, $\alpha^{T,j}$= 0.7.

5.5 Fault with Drifting Parameter

In this case, α_k^j is drifting after the fault. It is sampled uniformly from theinterval [0, $\alpha^{T,j}$] at the time of the fault occurrence and then follows a randomwalk (but bounded within [0, $\alpha^{T,j}$]), that is

$$\alpha_k^j = \begin{cases} \alpha^{T,j}, & if\ \alpha_{k|k-1}^j > \alpha^{T,j} \\ 0, & if\ \alpha_{k|k-1}^j < 0 \\ \alpha_{k|k-1}^j, & else \end{cases}$$

where

$$\alpha_{k|k-1}^j = \alpha_{k-1}^j + \ell_k$$

and ℓ_k is uniformly distributed random samples (i.e., $\ell_k \sim \mathcal{U}([-\sigma,\ \sigma]$. Asmentioned before, a sensor fault is easier to be detected. It is identified (almost) immediately when it happens. So, in this case we only evaluate the performancefor actuator faults. The results are given in Table 4. Compared with Case 2, the drifting α_k^j decreases the estimation accuracy and miss detection rate for both actuators, but increases the false identification rate significantly for actuator 1.

Fault	Algo.	CI	Fa	FI	MD	AD	$\breve{\alpha}$
A1	MMA	0.710	0.04	0.188	0.062	14.8	0.220
	EMA	0.724	0.04	0.160	0.076	15.3	0.208
	IMM	0.681	0.04	0.188	0.091	16.9	0.225
A2	MMA	0.745	0.08	0.106	0.069	15.6	0.236
	EMA	0.755	0.08	0.094	0.071	16.3	0.231
	IMM	0.726	0.08	0.118	0.076	17.9	0.245

Table 4. Fault with Drifting Parameter,
where $\alpha^{T,j} = 0.7$ and $\sigma = 0.02$.

6. Conclusions

Applying the generalized SSPRT to a linear dynamic system for fault isolation and estimation leads to the multiple model algorithm. However, this algorithm must be approximated in real applications due to the increasing computational demands and the unknown fault parameters. The Gaussian mixture reduction (GMR) and model augmentations are proposed to address these two problems, respectively.

The GMR reduces the number of Gaussian components in a greedy manner by merging iteratively components that are “close” to each other. This merging algorithm is based on more solid ground than the conventional GPBn method for multiple model algorithms. Further, the fault parameters can be estimated based on the augmented models.

The model augmentations by expectation and MLE are good options. They have their pros and cons as indicated by the simulation results. The MLE provides a shorter fault isolation delay and an smaller miss detection while the EMA performs better in terms of the correct isolation rate and the estimation accuracy for the unknown parameter.

As mentioned before, the isolation and estimation of a fault provide a reference for further actions, and hence infrequent sequential faults can also be dealt with. The case of simultaneous faults is more complicated and is considered as future work.

Acknowledgments

Research was supported by the Fundamental Research Funds for the Central Universities (China), project No. 12QN41.

References

[1] D. Middleton and R. Esposito, "Simultaneous optimum detection and estimation of signals in noise," IEEE Transactions on Information Theory, vol. 14, no. 3, pp. 434–444, May 1968.

[2] H. L. V. Trees, Detection, Estimation, and Modulation Theory, Part I: Detection, Estimation, and Linear Modulation Theory. New York: Wiley, 1971.

[3] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," Automatica, vol. 12, no. 6, pp. 601–611, November 1976.

[4] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy — asurvey and some new results," Automatica, vol. 26, no. 3, pp. 459–474, May 1990.

[5] M. Basseville and I. Nikiforov, Detection of Abrupt Changes: Theory and Application. Englewood Cliffs, NJ: Prentice Hall, 1993.

[6] I. V. Nikiforov, "A generalized change detection problem," IEEE Journal of Oceanic Engineering Transactions on Information Theory, vol. 41, no. 1, pp. 171–187, January 1995.

[7] Y. Zhang and X. R. Li, "Detection and diagnosis of sensor and actuator failures using IMM estimator," IEEE Transactions on Aerospace and Electronic Systems, vol. 34, no. 4, pp. 1293–1313, October 1998.

[8] D. P. Malladi and J. L. Speyer, "A generalized Shiryayev sequential probability ratio test for change detection and isolation," IEEE Transactions on Automatic Control, vol. 44, no. 8, pp. 1522–1534, August 1999.

[9] T. L. Lai, "Sequential multiple hypothesis testing and efficient fault detection-isolation in stochastic systems," IEEE Transactions on Information Theory, vol. 46, no. 2, pp. 595–608, March 2000.

[10] J. Ru and X. R. Li, "Variable-structure multiple-model approach to fault detection, identification, and estimation," IEEE Transactions on Control Systems Technology, vol. 16, no. 5, pp. 1029–1038, September 2008.

[11] E. S. Page, "Continuous inspection schemes," Biometrika, vol. 41, pp. 100–115, 1954.

[12] G. Lorden, "Procedures for reacting to a change in distribution," The Annals of Mathematical Statistics, vol. 42, no. 6, pp. 1897–1908, 1971.

[13] G. V. Moustakides, "Optimal stopping times for detecting changes in distributions," The Annals of Statistic, vol. 14, no. 4, pp. 1379–1387, December1986.

[14] A. N. Shiryayev, Optimal Stopping Rules. New York: Springer-Verlag, 1977.

[15] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking — part V: multiple-model methods," IEEE Transaction on Aerospace and Electronic Systems, vol. 41, no. 4, pp. 1255–1321, Oct. 2005.

[16] P. S. Maybeck and R. D. Stevens, "Reconfigurable flight control via multiple model adaptive control method," IEEE Transactions on Aerospace and Electronic Systems, vol. 27, no. 3, pp. 470–480, May 1991.

[17] P. S. Maybeck, "Application of multiple model adaptive algorithms to reconfigurable flight control," Control and Dynamic Systems, vol. 52, pp.291–320, 1992.

[18] T. E. Menke and P. S. Maybeck, "Sensor/actuator failure detection in the Vista F-16 by multiple model adaptive estimation," IEEE Transactions on Aerospace and Electronic Systems, vol. 31, no. 4, pp. 1218–1229, October1995.

[19] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," IEEE Transactions on Automatic Control, vol. 33, no. 8, pp. 780–783, August 1988.

[20] J. Ru and X. R. Li, "Interacting multiple model algorithm with maximum likelihood estimation for FDI," in IEEE International Symposium on Intelligent Control, Houston, TX, October 2003, pp. 661–666.

[21] J. F. Ru, "Adaptive estimation and detection techniques with applications," Ph.D. dissertation, University of New Orleans, New Orleans, LA, August 2005.

[22] J.-F. Ru, X. R. Li, and V. P. Jilkov, "Multiple model detection of target maneuvers," in Proceedings of Signal and Data Processing of Small Target, vol. 5913, 2005, pp. 100–108.

[23] J. L. Williams and P. S. Mayback, "Cost-function-based Gaussian mixture reduction for target tracking," in Proceedings of the 6th International Conference on Information Fusion, Cairns, Australia, July 2003.

[24] J. L. Williams, “Gaussian mixture reduction for tracking multiple maneuvering targets in clutter,” M.S.E.E, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, 2003.

[25] X. R. Li, V. P. Jilkov, and J. Ru, “Multiple-model estimation with variable structure - part VI: expected-mode augmentation,” IEEE Transactions on Aerospace and Electronic Systems, vol. 41, no. 3, pp. 853–867, July 2005.

[26] R. E. Kalman, “A new approach to linear filtering and prediction problems,” Journal of Basic Engineering, vol. 82, no. 1, pp. 35–45, 1960.

[27] D. J. Salmond, “Mixture reduction algorithms for target tracking in clutter,” in Proceedings of SPIE Signal and Data Processing of Small Targets, vol. 1305, October 1990, pp. 434–445.

[28] M. West, “Approximating posterior distributions by mixtures,” Journal of the Royal Statistical Society Series B Methodological, vol. 55, no. 2, pp. 409–422, 1993.

[29] P. S. Mayback and B. D. Smith, “Multiple model tracker based on Gaussian mixture reduction for maneuvering targets in clutter,” in Proceedings of 8th International Conference on Information Fusion, vol. 1, 2005, pp. 40–47.

[30] A. R. Runnalls, “Kullback-Leibler approach to Gaussian mixture reduction,” IEEE Transactions on Aerospace and Electronic Systems, vol. 43,no. 3, pp. 989–999, July 2007.

[31] M. Huber, P. Krauthausen, and U. D. Hanebeck, “Superficial Gaussian mixture reduction,” in Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion Trends Solutions Applications SDF, 2011.

[32] J. Zhang and Y. Liu, “Single maneuvering target tracking in clutter based on multiple model algorithm with Gaussian mixture reduction,” Journal of Applied Research and Technology, vol. 11, no. 5, pp. 641–652, 2013.

[33] J. R. Hershey and P. A. Olsen, “Approximating the Kullback-Leibler divergence between Gaussian mixture models,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Honolulu, HI, April 2007, pp. IV–317 – IV–320.

[34] Y. Zhu, “Efficient recursive state estimator for dynamic systems without knowledge of noise covariances,” IEEE Transactions on Aerospace and Electronic Systems, vol. 35, no. 1, pp. 102–114, January 1999.

[35] Y. Zhu, K. Zhang, and X. R. Li, “Fusion of distributed extended forgetting factor RLS state estimators,” IEEE Transactions on Aerospace and Electronic Systems, vol. 44, no. 2, pp. 457–467, April 2008.