International Journal of Innovative Technology and Interdisciplinary Sciences
www.IJITIS.org
ISSN: 2613-7305
Volume 2, Issue 2, pp. 160-172, 2019
DOI: https://doi.org/10.15157/IJITIS.2019.2.2.160-172
Received August 20, 2018; Accepted November 28, 2018

New Approach for a Fault Detect Model-Based Controller[i]

Nitin Afzulpurkar

Mechatronics, SAT/ISE
Asian Institute of Technology, Bangkok, Thailand
nitin@ait.ac.th

ABSTRACT

In this paper, we present a design of a new fault detect model-based (FDMB) controller system. The system is aimed to detect faults quickly and reconfigure the controller accordingly. Thus, such system can perform its function correctly even in the presence of internal faults. An FDMB controller consists of two main parts, the first is fault detection and diagnosis (FDD); and the second is controller reconfiguration (CR). Systems subject to such faults are modelled as stochastic hybrid dynamic model. Each fault is deterministically represented by a mode in a discrete set of models. The FDD is used with interacting multiple-model (IMM) estimator and the CR is used with generalized predictive control (GPC) algorithm. Simulations for the proposed controller are illustrated and analysed.

Keywords: Fault Detect Model-Based, Fault Detection and Diagnosis, Controller Reconfiguration, Interacting Multiple-Model, Generalized Predictive Control.

1. INTRODUCTION

This paper concerns the development of a controller to deal with faults in sensors, actuators and components. In reality, an early and robust fault detection is difficult to achieve amid the presence of disturbances and measurement noises as well as uncertainties. We therefore examine and propose a fault detect model-based (FDMB) controller system for fault detection and diagnosis (FDD) and controller reconfiguration (CR). The outline for this paper is as follows: Section 2 describes the design and verification of fault modelling; Section 3 analyses the selection of an FDD system; Section 4 develops a CR in an integrated FDMB controller; Examples and simulations are given after each section to illustrate the main ideas in the section; finally conclusions are given in section 5.

2. FAULT MODELLING

Faults are difficult to foresee and prevent. Traditionally faults were handled by describing the resulting behaviour of the system and grouped into a hierarchic structure of fault models [1]. This traditional approach is still widely used in practice and a number of methods for fault modelling and detection of systems have been developed. When a failure occurs, the system behaviour changes and should be described by a

---

[i] *This paper is a revised and expanded version of a paper entitled 'Development of a Fault Detection Model-Based Controller' presented at the Third International Conference on High Performance Scientific Computing, March 6–10, 2016, Hanoi, Vietnam.*

different mode from the one that corresponds to the normal mode of operation. In the traditional methods, the state of failure in each mode is fixed and cannot jump randomly among modes.

For dynamic systems in which state may jump as well vary continuously in a discrete set of modes, an effective way to model the faults is so-called stochastic hybrid system. Apart from the applications to problems involving failures, hybrid systems have found great success in other areas such as target tracking and control that involves possible structural changes [2]. The stochastic hybrid model assumes that the actual system at any time can be modelled sufficiently and accurately by:

$$x(k+1) = A(k, m(k+1))x(k) + B(k, m(k+1))u(k) + T(k, m(k+1))\xi(k, m(k+1)), \tag{1}$$

$$z(k) = C(k, m(k+1))x(k) + \eta(k, m(k+1)), \tag{2}$$

with the system mode sequence assumed to be a first-order Markov chain with transition probabilities:

$$P\{m_j(k+1) \mid m_i(k)\} = \pi_{ij}(k), \quad \forall m_i, m_j \in \mathcal{I} \tag{3}$$

where $x \in \mathbb{R}^n$ is the state vector; $z \in \mathbb{R}^p$ is the measured output; $u \in \mathbb{R}^m$ is the control input; $\xi \in \mathbb{R}^{n_\xi}$ and $\eta \in \mathbb{R}^p$ are independent discrete-time response and measurement noises with mean $\bar{\xi}(k)$ and $\bar{\eta}(k)$, and covariance $Q(k)$ and $R(k)$; $P\{.\}$ denotes probability; $m(k)$ is the discrete-valued modal state, i.e. the index of the normal or fault mode, at time *k*, which denotes the mode in effect during the sampling period ending at *k*; $\mathcal{I} = \{m_1, m_2, ..., m_N\}$ is the set of all possible system modes; $\pi_{ij}(k)$ is the transition probability from mode m_i to mode m_j, i.e. the probability that the system will jump to mode m_j at time instant *k*. Obviously, the following relation must be held for any $m_i \in M$:

$$\sum_{j=1}^{N} \pi_{ij}(k) = \sum_{j=1}^{N} P\{m_j(k+1) \mid m_i(k)\} = 1, \quad i = 1, ..., N \tag{4}$$

Fault in sensors or actuators can be modelled by changing the appropriate column (or row) of matrix B or C in equation (1) or (2) by a scaling factor that represents the effectiveness of sensors or actuators failures in the system. They can also be modelled by increasing the process noise covariance matrix *Q* or measurement noise covariance matrix R in ξ, and η:

$$M_i \in M = \begin{cases} x(k+1) = A(k)x(k) + B_i(k)u(k) + T_i(k)\xi_i(k) \\ z(k) = C_i(k)x(k) + \eta_i(k) \end{cases} \tag{5}$$

where the subscript *i* denotes the fault modelling in model set $M_i \in M = \{M_1, ..., M_N\}$, each M_i corresponds to a node (a fault) occurring in sensors or actuators system.

We then now verify and check the distances between models in the model set *M*. If the distance between two models is short, they might not be identifiable by the FDD. In this paper, the distance between two models is calculated via the difference of their H_∞ norm, i.e. the distance *d* between two models M_1 and M_2 is defined as:

$$d = \|M_1 - M_2\|_\infty \tag{6}$$

Example 1: Fault Model Set Design and Verification

Simulations throughout of this paper are used the following process model of a distillation column with four state variables, one input (feed flow rate) and two outputs (overhead flow rate and overhead composition). The state space model of the system is:

$$M = \begin{cases} \dot{x}(t) = \begin{bmatrix} -0.05 & -6 & 0 & 0 \\ -0.01 & -0.15 & 0 & 0 \\ 1 & 0 & 0 & 13 \\ 0 & 1 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} -0.2 \\ 0.03 \\ 2 \\ 0.1 \end{bmatrix} u(t) + \xi_x(t) \\ z(t) = \begin{bmatrix} 1 & -0.5 & 1 & 1 \\ -1 & 0.6 & 0 & 1 \end{bmatrix} x(t) + \xi_z(t) \end{cases} \tag{7}$$

where x is the state variable, u is the input and z is the measurement outputs, ξ_x *and* ξ_z *are assumed as white noises with covariance* $Q = 0.01$ *for process and* $R = 0.01$ *for measurement, respectively. The model is discretized with sampling period T=0.1 sec.*

The following five models are including in the model set:

- *Model 1: Nominal model (no fault), then nothing changes in equation (7),*
- *Model 2: Total sensor 1 failure, then* $z_{M2}(t) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 0.6 & 0 & 1 \end{bmatrix} x(t) + \xi_z(t)$,
- *Model 3: Total sensor 2 failure, then* $z_{M3}(t) = \begin{bmatrix} 1 & -0.5 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} x(t) + \xi_z(t)$,
- *Model 4: Sensor1 failure 50%, then* $z_{M4}(t) = \begin{bmatrix} 0.5 & -0.25 & 0.5 & 0.5 \\ -1 & 0.6 & 0 & 1 \end{bmatrix} x(t) + \xi_z(t)$,
- *Model 5: Sensor 2 failure 50%, then* $z_{M5}(t) = \begin{bmatrix} 1 & -0.5 & 1 & 1 \\ -0.5 & 0.3 & 0 & 0.5 \end{bmatrix} x(t) + \xi_z(t)$.

Now, we check the distances between models (Table 1):

Table 1. Distances between models

Models	M_1	M_2	M_3	M_4	M_5
M_1	*0*	*479*	*85*	*239*	*322*
M_2	*479*	*0*	*486*	*239*	*578*
M_3	*85*	*486*	*0*	*254*	*239*
M_4	*239*	*239*	*254*	*0*	*401*
M_5	*322*	*578*	*239*	*01*	*0*

3. FAULT DETECTION AND DIAGNOSIS (FDD)

In this section, we analyse and select a fast and reliable FDD system applied for the above set of models by using algorithms of multiple model (MM) estimators. MM estimation algorithms appeared in early 1980s when Shalom and Tse [5] introduced a suboptimal, computationally-bounded extension of Kalman filter to cases where measurements were not always available. Then, several multiple-model filtering techniques, which could provide accurate state estimation, have been developed. Major existing approaches for MM estimation are discussed and introduced in ([4],[5],[6],[7],[8]) including the Non-Interacting Multiple Model (NIMM), the Gaussian Pseudo Bayesian (GPB1), the Second-order Gaussian Pseudo Bayesian (GPB2), and the Interacting Multiple Model (IMM).

From the design of model set, a bank of filters runs in parallel at every time, each based on a particular model, to obtain the model-conditional estimates. The overall state estimate is a probabilistically weighted sum of these model-conditional estimates. The jumps in system modes can be modelled as switching between the assumed models in the set.

Figure 2 shows the operation of the recursive multiple model estimator, where $\hat{x}_i(k|k)$ is the estimate of the state $x(k)$ obtained from the filter based on model M_i at time k given the measurement sequence through time k; $\hat{x}_i^0(k-1|k-1)$ is the equivalent reinitialized estimate at time (k-1) as the input to the filter based on model M_i at time k; $\hat{x}(k|k)$ is the overall state estimate; $P_i(k|k)$, $P_i^0(k-1|k-1)$ and $P(k|k)$ are the corresponding covariances.

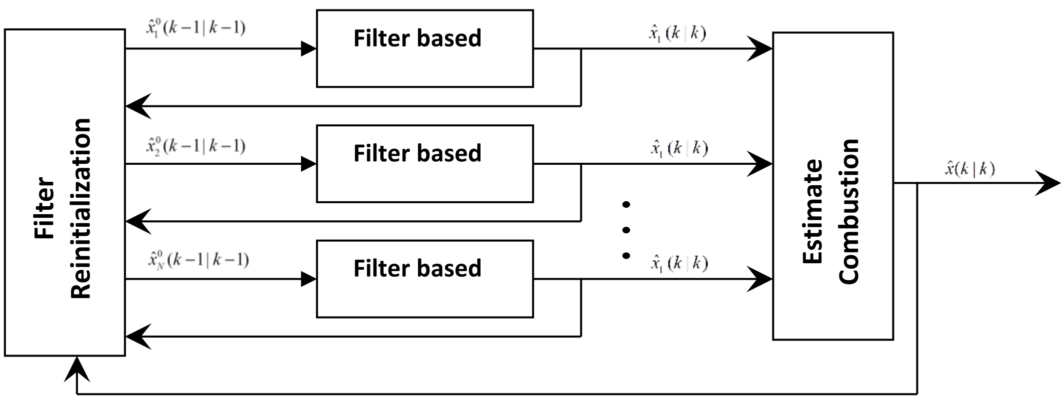

Figure 2. Structure of a MM estimator.

A simple and straightforward way of filter reinitialization is: Each single model based recursive filter uses its own previous state estimation and state covariance as the input at the current cycle:

$$\begin{aligned} \hat{x}_i^0(k-1|k-1) &= \hat{x}_i(k-1|k-1) \\ P_i^0(k-1|k-1) &= P_i(k-1|k-1) \end{aligned} \tag{8}$$

This leads to the so-called non-interacting multiple model (NIMM) estimator because the filters operate in parallel without interactions with one another, which is reasonable only under the assumption that the system mode does not change (Figure 3).

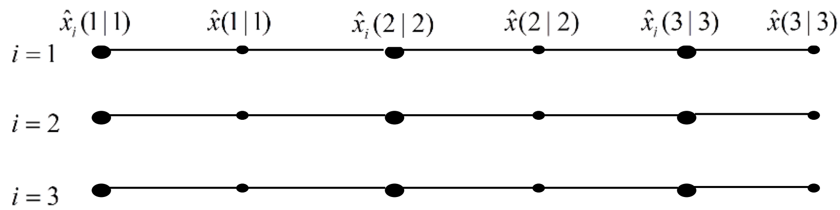

Figure 3. Illustration of the NIMM estimator

Another way of reinitialization is to use the previous overstate estimate and covariance for each filter as the required input:

$$\begin{aligned} \hat{x}_i^0(k-1|k-1) &= \hat{x}(k-1|k-1) \\ P_i^0(k-1|k-1) &= P(k-1|k-1) \end{aligned} \tag{9}$$

This leads to the first order Generalized Pseudo Bayensian (GPB1) estimator (Figure 4). It belongs to the class of interacting multiple model estimators since it uses the previous overall state estimate, which carries information from all filters. Clearly, if the transition probability matrix is an identity matrix this method of reinitialization reduces to the first one.

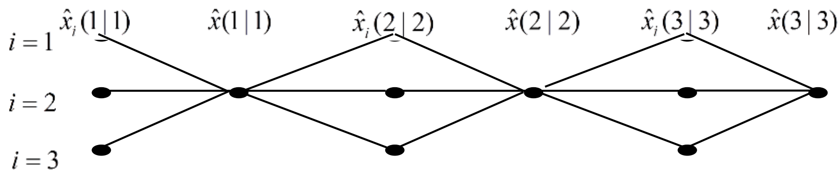

Figure 4. Illustration of the GPB1 estimator

The GPB1 and GPB2 algorithms were the result of early work by Ackerson and Fu [6] and good overviews are provided in [7], where suboptimal hypothesis pruning techniques are compared. The GPB2 differed from the GPB1 by including knowledge of the previous time step's possible mode transitions, as modelled by a Markov chain. Thus, GPB2 produced slightly smaller tracking errors than GPB1 during non-manoeuvring motion. However in the size of this paper, we do not include GPB2 into our simulation test and comparison.

A significantly better way of reinitialization is to use IMM. The IMM was introduced by Blom in [8] and Zhang and Li in [4]:

$$\begin{aligned} \hat{x}_j^0(k|k) &= E[x(k|z^k, m_j(k+1)] = \sum_i \hat{x}_i(k:k)P\{m_i(k)|z^k, m_j(k+1)\} \\ P_i^0(k|k) &= \operatorname{cov}[\hat{x}_j^0(k:k)] = \sum_i P\{m_i(k)|z^k, m_j(k+1)\times\{P_i(k|k)+\tilde{x}_{ij}^0(k|k)\tilde{x}_{ij}^0(k|k)' \end{aligned} \tag{10}$$

where $\operatorname{cov}[.]$ stands for covariance and $\tilde{x}_{ij}^0(k|k)=\hat{x}_i^0(k|k)-\hat{x}_j(k|k)$. Figure 5 depicts the reinitialization in the IMM estimator. In this paper we use this approach for setting up a FDD system.

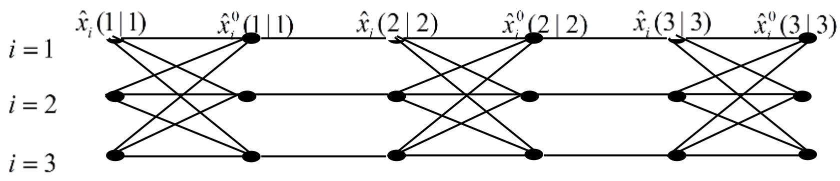

Figure 5. Illustration of the IMM estimator

For each model, we can operate a Kalman filter. The probability of each model matching to the system mode provides the required information for mode chosen decision. The mode decision can be achieved by comparing with a fixed threshold probability μ_T. If the mode probabilities $\max_i(\mu_i(k)) \geq \mu_T$, mode at $\mu_i(k)$ is occurred and taken place at the next cycle. Otherwise, there is no new mode detection. The system maintains the current mode for the next cycle calculation.

Example 2: Analysis and Selection of FDD system

From the fault model set design in Example 1. We assume the model mode can jump from each other in a mode probability matrix as:

$$\pi_{ij} = \begin{bmatrix} 0.96 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.05 & 0.95 & 0 & 0 & 0 \\ 0.05 & 0 & 0.95 & 0 & 0 \\ 0.05 & 0 & 0 & 0.95 & 0 \\ 0.05 & 0 & 0 & 0 & 0.95 \end{bmatrix}$$

The threshold value for the mode probabilities is chosen $\mu_T = 0.9$. *Now we begin to compare the three estimators of NIMM, GPB1, and IMM to test their ability to detect faults. The five models are run with time:* M_1 *for k=1-20, k=41-60, k=81-100, k=121-140, and k=161-180;* M_2 *for k=21-40;* M_3 *for k=61-80;* M_4 *for k=101-120; and* M_5 *for k=141-160. Results of simulation are shown in Figure 6.*

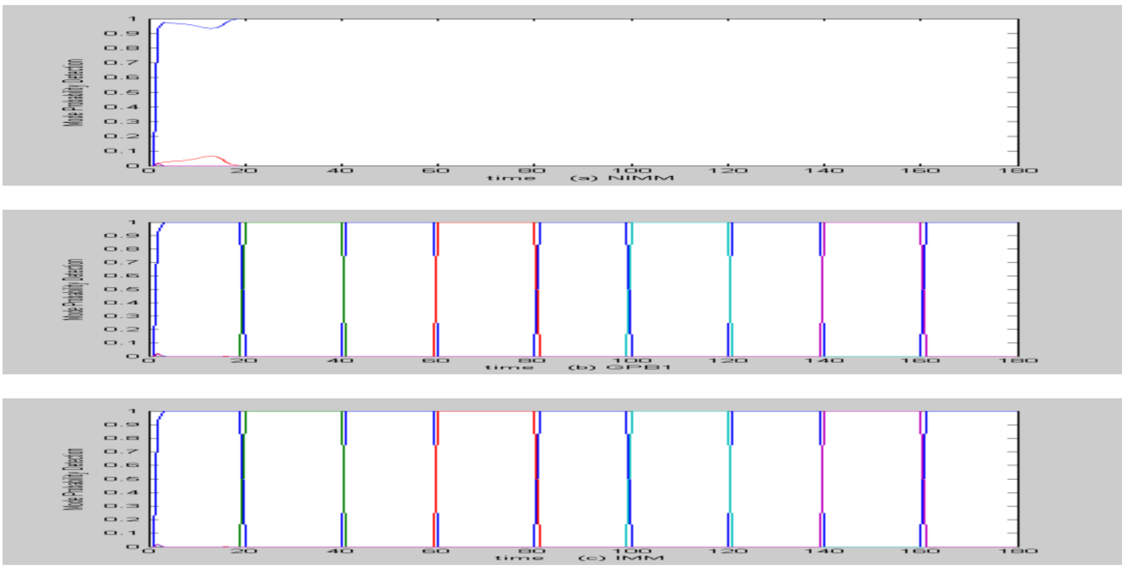

Figure 6. Comparison of probabilities of estimators (a) NIMM, (b) GPB1, and (c) IMM.

In Figure 6, we can see that the GPB1 estimator performs as good as IMM estimator while NIMM estimator fails to detect sensor failures in the model set.

Next we will test the ability of the GPB1 and IMM estimators to detect sensor failures by narrowing the distances between modes as close as possible until one of methods cannot detect the failures. Now we design the following mode set: $M_1 =$ *Model 1: Nominal mode;* $M_2 =$ *Model 2: Sensor 1 failure 5%;* $M_3 =$ *Model 3: Sensor 2 failure 5%;* $M_4 =$ *Model 4: Sensor 1 failure 2%; and* $M_5 =$ *Model 5: Sensor 2 failure 2%. With these parameters, we achieve the following distances between the closed-loop systems (Table 3).*

Table 3. Distances between the closed-loop systems

Models	M_1	M_2	M_3	M_4	M_5
M_1	0	0.0480	0.0085	0.2400	0.0043
M_2	0.0480	0	0.0487	0.1920	0.0482
M_3	0.0085	0.0487	0	0.2401	0.0043
M_4	0.2400	0.1920	0.2401	0	0.2400
M_5	0.0043	0.0482	0.0043	0.2400	0

Since the distances between models are very close and the GPB1 fails to detect failures while IMM still proves it's much superior capability to detect failures over GPB1 (Figure 7).

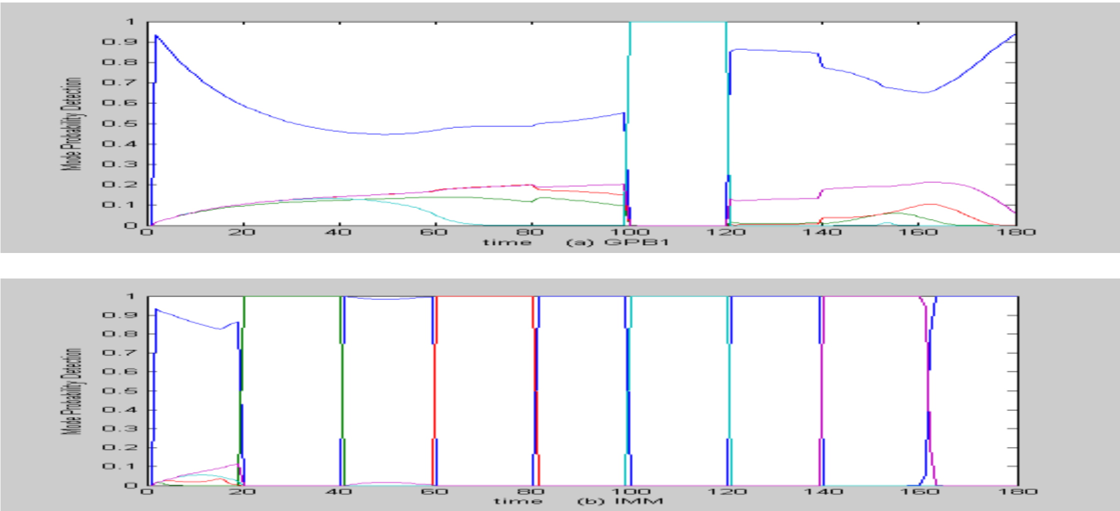

Figure 7. Comparison of probabilities of estimators (a) GPB1, and (b) IMM

As a result, we select the IMM for our FDD system. Now we move to the next step to design a suitable controller reconfiguration for the FDMB system.

4. CONTROLLER RECONFIGURATION (CR)

In this section we develop a new CR which can online determine the optimal control actions and reconfigure the controller using Generalized Predictive Control (GPC) algorithm. We will show how an IMM-based GMC controller can be a good FDMB system.

Firstly we review the basic GPC algorithm: Generalized Predictive Control (GPC) is one of MPC techniques developed by Clarke et al. ([9],[10]): GPC was intended to offer a new adaptive control alternative. GPC uses the ideas with controlled autoregressive integrated moving average (CARIMA) plant in adaptive context and self-tuning by recursive estimation. Kinnaert [11] developed GPC from CARIMA model into a more general in state-space form.

Consider the state space of the stochastic model in the innovation form [11]:

$$\bar{M}:\begin{cases}\hat{x}(k+1\,|\,k)=\tilde{A}\hat{x}(k\,|\,k-1)+\tilde{B}\Delta u(k)+\tilde{K}e(k)\\ z(k)=\tilde{C}\hat{x}(k\,|\,k-1)+e(k)\end{cases}\tag{11}$$

where $e(k)$ is an innovation sequence, $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, and $\tilde{K}$ are fixed matrices, $z(k)\in\mathbb{R}^{n_y}$, $\Delta u(k)\in\mathbb{R}^{n_u}$, and $\hat{x}(k\,|\,k-1)$ is an estimate of the state $x(k)$ obtain from a Kalman filter.

For a moving horizon control with the augmented system in equation (11), the prediction of $x(k+j\,|\,k)$ given the information $(z(k),z(k-1),...;u(k-1),u(k-2),...)$ is:

$$\hat{x}(k+j\,|\,k)=A^{j}\hat{x}(k\,|\,k-1)+\sum_{i=0}^{j-1}A^{j-1-i}B\Delta u(k+i)+A^{j-1}Ke(k)\tag{12}$$

and the prediction of the filtered output will be:

$$\hat{z}(k+j\,|\,k)=CA^{j}\hat{x}(k\,|\,k-1)+\sum_{i=0}^{j-1}CA^{j-1-i}B\Delta u(k+i)+CA^{j-1}Ke(k)\tag{13}$$

If we form:

$\tilde{u}(k)=\left[\Delta u^{'}(k),\ ...\ ,\ \Delta u^{'}(k+NU-1)\right]^{'}$ and $\tilde{z}(k)=\left[\hat{z}^{'}(k+N_1\,|\,k),\ ...\ ,\ \hat{z}^{'}(k+N_2-1\,|\,k)\right]^{'}$, we can write the global predictive model for the filtered out for from N_1 to N_2 output prediction and for from *1* to NU input prediction horizons as:

$$\tilde{z}(k)=\begin{bmatrix}CB & 0 & \cdots & 0\\ CAB & CB & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ CA^{NU-1}B & CA^{NU-2}B & \cdots & CB\\ \vdots & \vdots & \vdots & \vdots\\ CA^{N_2-1}B & CA^{N_2-2}B & \cdots & CA^{N_2-NU}B\end{bmatrix}\tilde{u}(k)+\begin{bmatrix}CA\\ CA^2\\ \vdots\\ \vdots\\ \vdots\\ CA^{N_2}\end{bmatrix}\hat{x}(k\,|\,k-1)+\begin{bmatrix}C\\ CA\\ \vdots\\ \vdots\\ \vdots\\ CA^{N_2-1}\end{bmatrix}Ke(k)$$

For simplicity, we can rewrite as:

$$\tilde{z}(k)=U\tilde{u}(k)+V\hat{x}(k\,|\,k-1)+WKe(k)\tag{14}$$

Consider the general cost function in GPC

$J=\sum_{j=N_1}^{N_2}\left\|z(k+j)-w(k+j)\right\|+\sum_{j=1}^{NU}\left\|\Delta u(k+j-1\right\|_{\Gamma}$ with $w(k+j)$ is the output reference and Γ is the control weighting matrix, the control law that minimizes this cost function is:

$$\tilde{u}(k) = -(U'U + \Gamma)^{-1}U' \left(V\hat{x}(k \mid k-1) + WKe(k) - w(k)\right) \tag{15}$$

then the first input $\Delta u(1)$ in $\tilde{u}(k)$ will be implemented into the system.

Now, we can combine GPC with IMM estimator: Since GPC follows a stochastic perspective, we can use GPC controller for the CR using the inputs of the CR as the outputs of the IMM. The overall state estimate $x(k) \approx \hat{x}(k) = \sum_{i=1}^{N} \mu_i(k)\hat{x}_i(k)$ where N is the number of models in the model set. So we can assume that the "true" system is the weighted sum with $\mu_i(k)$ of models in a convex combination $M = \{M_1, \ldots, M_N\}$.

A generalized diagram of IMM based GPC controller is shown in Figure 8.

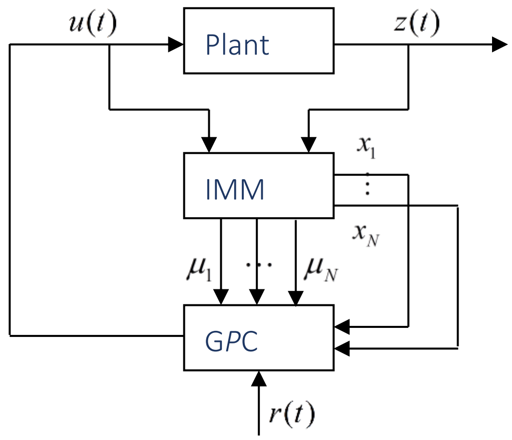

Figure 8. Diagram of IMM based GPC controller

We build a bank of GPC controllers for each model in the model set. Assuming the mode probabilities $\mu_i(k)$ are constant during the control horizon, we can easily derive the GPC control law by forming $U = \left(\sum_{i=1}^{N} \mu_i U_i\right)$, $V = \left(\sum_{i=1}^{N} \mu_i V_i\right)$ and $W = \left(\sum_{i=1}^{N} \mu_i W_i\right)$ matrices that correspond to the "true" model $M = \sum_{i=1}^{N} \mu_i M_i$ in equation (14) and find out the optimal control action in equation (15). Then the first input $\Delta u(1)$ in $\tilde{u}(k)$ will be implemented into the system.

A notation is taken for one of disadvantages of the IMM-based GPC controller is that the type and the magnitude of the input excitation play an important role in its performance. When the magnitude of the input signal is very small, the residuals of the Kalman filters will be very small and, therefore, the likelihood functions for the modes will approximately be equal. This will lead to unchanging (or very slowly changing) mode probabilities which in turn makes the IMM estimator incapable to detect failures. Latest studies on the issue are referred to in [12], [13], [14], [15], and [16].

Example 3: Controller Reconfiguration (CR) and a new integrated FDMB system

Example 3.1. We run a normal GPC controller with $N_1 = 1$, $N_2 = 4$, $NU = 4$, *the weighting matrix* Γ *=0.1 and with a reference set-point at* $r = 1$. *From time k=51-100, Sensor 1 failure 50% of the real measurement. Continuously, from k=101-150, Sensor 1 failure to 150% the real measurement. That leads the normal GPC controller providing the wrong output (Figure 10).*

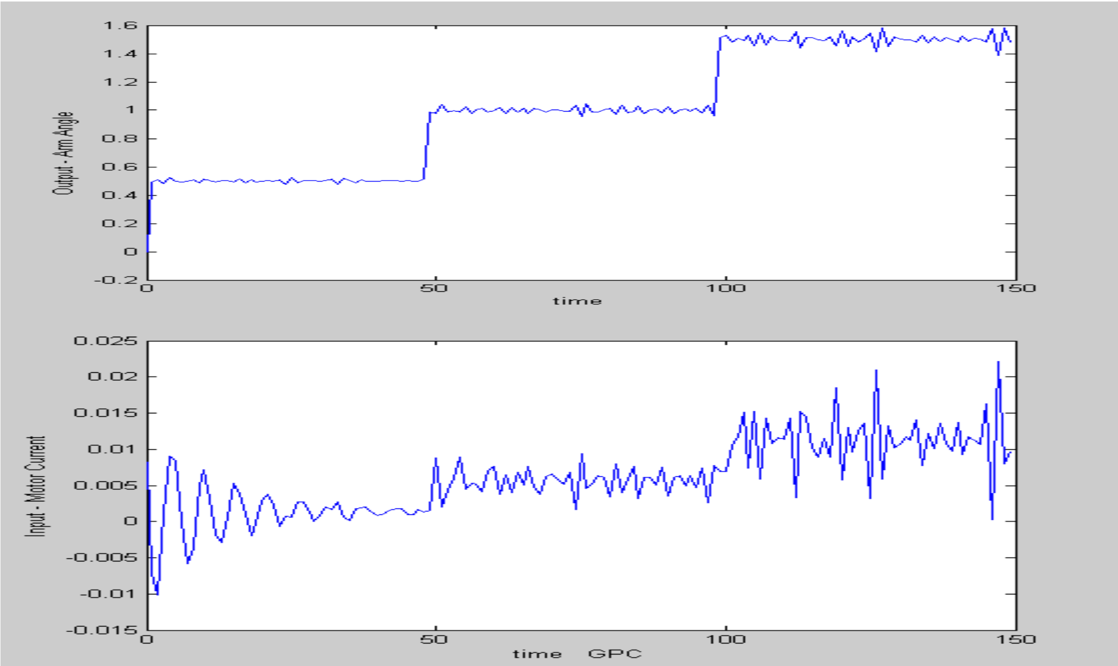

Figure 10. GPC controller with sensor errors (a) Output and (b) Input

Example 3.2. We run the same parameters in Example 3.1 using our proposed IMM-base GPC controller with the transition matrix $\pi_{ij} = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix}$. *Results are shown in Figure 11.*

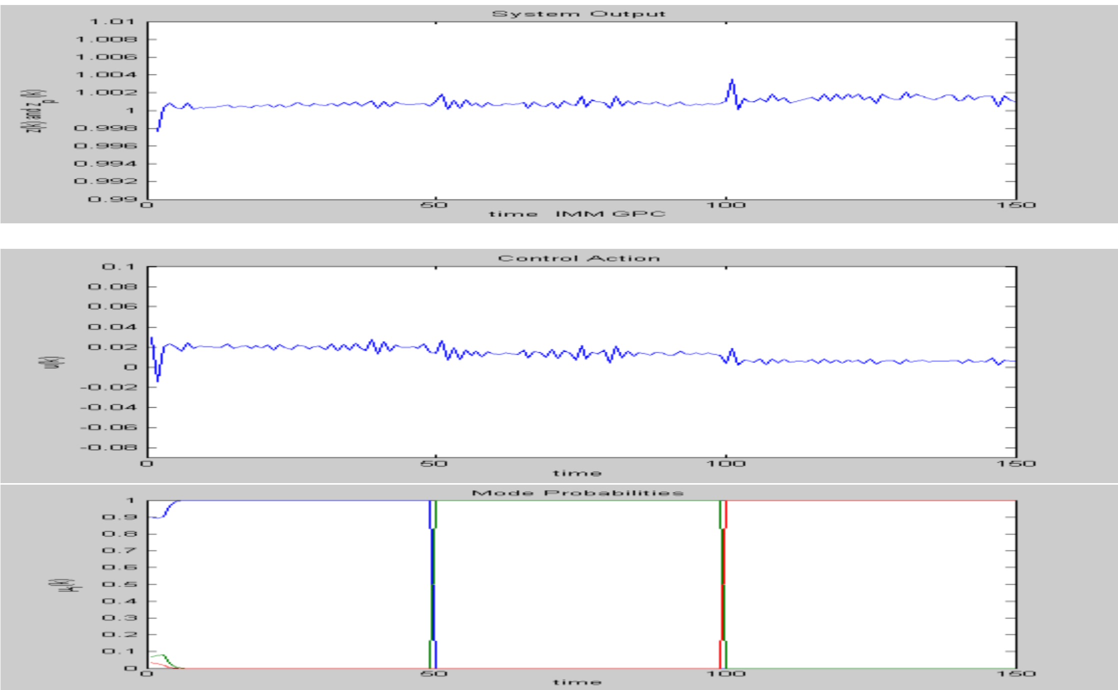

Figure 11. IMM-based GPC Controller (a) Output, (b) Input, and (c) Probabilities

Our new FDMB system still keeps the output at the desired setpoint since the IMM estimator easily finds accurate fault mode and activate the CR system online. We can see two output jumps at time 61 and 101 because the new mode appears due to the activation of the sensor fault mode. However the output returns back the setpoint immediately because our FDMB system has reconfigured the controller upon the fault mode changes.

For demonstrating the disadvantages of our FDMB system due to the low magnitude of input signals, we run the same parameters in Example 3.2 but reduce the reference setpoint to a very low value at $r = 0.01$. *Our IMM-based GPC controller fails to detect the sensor error at time 61 and time 101. The system becomes uncontrollable (Figure 12).*

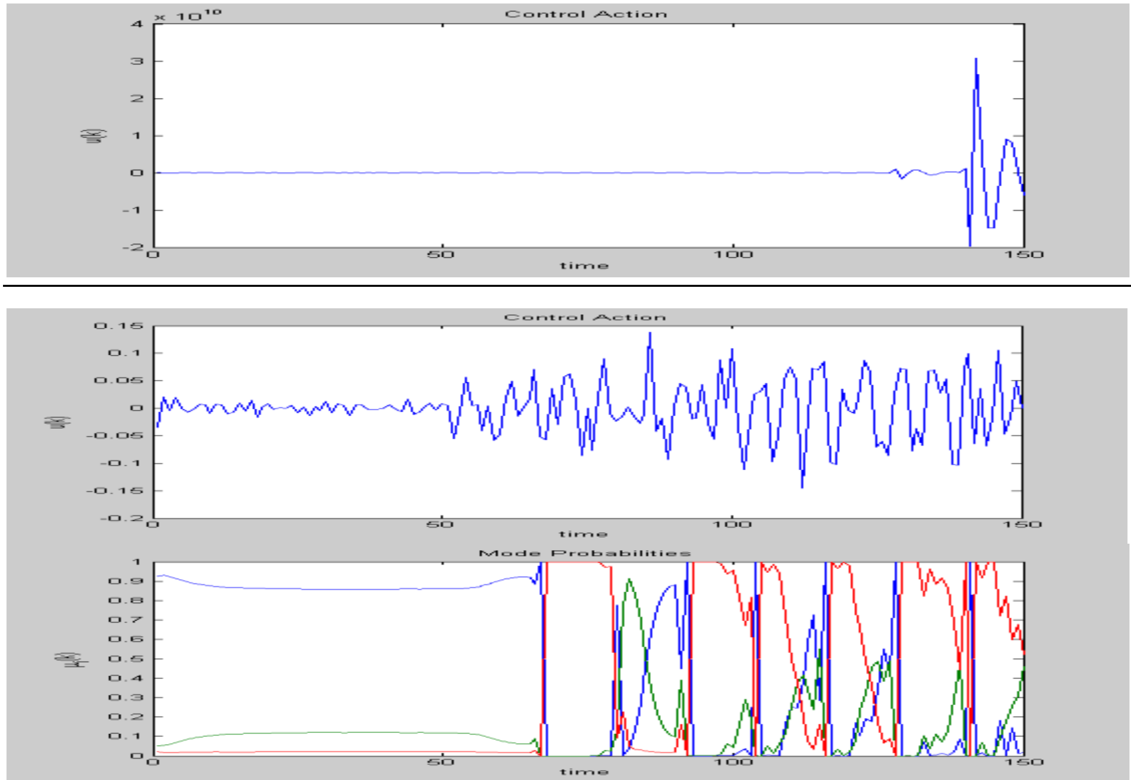

Figure 12. IMM-based GPC failure with very low magnitude of input signal

(a) Output, (b) Input, and (c) Probabilities

5. CONCLUSION

Systems subject to sensor and activator failures can be modelled as a stochastic hybrid system with fault modelling nodes in the model set. IMM based GPC controller can provide real-time control performance and detection and diagnosis of sensor and actuator failures online. Simulations in this study show that IMM estimator is superior than NIMM and GPB1 estimators. Our proposed IMM based GPC controller also proves its ability better than a normal GPC controller in maintaining the output setpoints amid sensor or actuator failures.

The main difficulty of this approach is the choice of modes in the model set as well as the transition probability matrix that assigns probabilities for jumps from one mode to another since the IMM algorithms are sensitive to the transition probability matrix. Another limitation related to IMM based GPC controller is the magnitude of the input excitation. When we change the output setpoint to small values, the input signal might become very small and this leads to unchanging mode probabilities or IMM based GPC controller cannot detect failures. Lastly, this approach does not consider issues of uncertainty in the controller system.

ACKNOWLEDGMENT

The author would like to thank Asian Institute of Technology (AIT) for supporting this research project.

CONFLICT OF INTERESTS

The author would like to confirm that there is no conflict of interests associated with this publication and there is no financial fund for this work that can affect the research outcomes.

REFERENCES

[1] Cristian, F., “Understanding Fault Tolerant Distributed Systems”. *Communications of ACM*, Vol. 34, pp. 56-78, 1991.

[2] Li, R., “Hybrid Estimation Techniques”, *Control and Dynamic Systems, New York, Academic Press,* Vol. 76, pp. 213-287, 1996.

[3] Kanev, S., Verhaegn, M., “Controller Reconfiguration for Non-linear Systems”, Control Engineering Practice, Vol 8, 11, pp. 1223-1235, 2000.

[4] Zhang, Y., Li, R., “Detection and Diagnosis of Sensor and Actuator Failures Using IMM Estimator”, *IEEE Trans. On Aerospace and Electronic Systems*, Vol. 34, 4, 1998.

[5] Shalom Y., and Tse, E., “Tracking in a Cluttered Environment with Probabilistic Data Association”, Automatica, Vol. 11, pp. 451-460, 1975.

[6] Ackerson, G., and Fu, K., “On State Estimation in Switching Environments”. IEEE Trans. On Automatic Control, Vol. 15, 1, pp. 10-17, 1970.

[7] Tugnait, J., “Detection and Estimation of Abruptly Changing Systems”. Automatica, Vol 18, pp. 607-615, 1982.

[8] Blom, H., and Shalom, Y., “The Interacting Multiple Model Algorithm for System with Markovian Switching Coefficients” IEEE Trans on Automatic Control, Vol 33, 8, pp. 780-785, 1983.

[9] Clarke D. W., C. Mohtadi and P. S. Tuffs (1987[b]). ‘Generalized Predictive Control – Extensions and Interpretations’ *Automatica*, 23(2), 149-160.

[10] Clarke, D.W. C. Mohtadi and P.S. Tuffs (1987[a]) “Generalized Predictive Control: I. The Basic Algorithm” *Automatica*, 23(2), 137-147.

[11] Kinnaert M. (1989). ‘Adaptive Generalized Predictive Control for MIMO Systems’ *Int. J. Control*. 50(1), 161-172.

[12] VT Minh, Mohd Hashim, ‘Adaptive teleoperation system with neural network-based multiple model control’, *Mathematical Problems in Engineering* Volume 2010, Article ID 592054, 15 pages, 2010. http://dx.doi.org/10.1155/2010/592054.

[13] VT Minh, John Pumwa, ‘Feasible path planning for autonomous vehicles’ *Mathematical Problems in Engineering* Volume 2014, Article ID 317494, 12 pages, 2014. http://dx.doi.org/10.1155/2014/317494.

[14] VT Minh, *et al*. ‘Development of a Wireless Sensor Network Combining MATLAB and Embedded Microcontrollers’, *Sensor Letters*, 13(12), pp. 1091-1096, 2015. http://dx.doi.org/10.1166/sl.2015.3594.

[15] V.T. Minh and R. Khanna, “Application of Artificial Intelligence in Smart Kitchen”, *International Journal of Innovative Technology and Interdisciplinary Sciences,* vol. 1, no. 1, pp. 1-8, Nov. 2018.

[16] I. Ovchinnikov and P. Kovalenko, “Predictive Control Model to Simulate Humanoid Gait”, *International Journal of Innovative Technology and Interdisciplinary Sciences*, vol. 1, no. 1, pp. 9-17, Nov. 2018.