Infotech@Aerospace 2011
29 - 31 March 2011, St. Louis, Missouri

AIAA 2011-1587

Conflict Resolution Algorithms For Fault Detection and Diagnosis

Ali Nasir* and Ella M. Atkins.†, Ilya V. Kolmanovsky‡
University of Michigan, Ann Arbor, Michigan, 48105

Abstract

We present two approaches for conflict resolution between two fault detection schemes, detecting the same fault, via optimization with bounded adjustment of detection thresholds. In our first method, we assume initially that there is no conflict and optimize the thresholds of both schemes with respect to a partial cost function that penalizes false alarms and missed detections. Then we continuously update thresholds based on a comprehensive cost function that penalizes conflicts in addition to false alarms and missed detections. Our updates are bounded and controlled in such a way that the cost function always assumes the lowest possible cost as a function of thresholds. We make use of residual signals to minimize computational complexity. In our second method, we present a more general solution to the conflict resolution problem using a Markov Decision Process framework that generates an optimal policy for fault detection threshold. This method is computationally more complex but it is more general, does not require knowledge of residuals, and does not require initial optimization of the thresholds. We introduce an error signal that indicates failure in resolving the conflict using threshold updating in which case, a supervisor (human or computer) can be alerted and prompted to take a corrective action. We implemented our methods on a spacecraft attitude control thruster-valve system simulation with high noise. Our results show good performance and substantial reduction in conflicts under highly uncertain conditions.

Nomenclature

a_+^i = Penalty weight for missed detection of fault by detection scheme *i*

a_-^i = Penalty weight for false alarm of fault by detection scheme *i*

b^i = Binary flag indicating presence of fault detected by scheme *i* (depends on thresholds and input to the fault detection scheme).

v^i = Threshold value for fault detection in scheme *i*

$\bar{v}^i$ = Optimal value of threshold based on receiver operating characteristics of detection scheme *i*

$\hat{v}^i$ = Upper bound on threshold value based on penalties in the cost function

$\check{v}^i$ = Lower bound on threshold value based on penalties in the cost function

q = Penalty on conflict between detection schemes

c = Binary flag representing presence of conflict

J = Cost function

$P(MD^i)$ = Probability of missed detection for scheme *i*

$P(FA^i)$ = Probability of false alarm for scheme *i*

r^i = Residual signal for scheme *i* indicating difference between the output and the threshold

α^i = Primary parameter indicating the optimal amount by which threshold should change to resolve the conflict

β^i = Secondary parameter indicating the optimal amount by which threshold should change to resolve the conflict

* Graduate Student, Aerospace Engineering, Ann Arbor, MI 48109, email: techibro@umich.edu, AIAA student member.
† Associate Professor, Aerospace Engineering, Ann Arbor, MI 48109, email: ematkins@umich.edu, Associate Fellow.
‡ Professor, Aerospace Engineering, Ann Arbor, MI 48109, email: ilya@umich.edu, AIAA member.

f^i = Binary flags indicating whether or not the threshold of scheme *i* must be changed to resolve a conflict.

σ^i = Change in the cost function due to change in threshold to resolve a conflict

e = Binary flag indicating failure of threshold change in resolving the conflict

κ = Binary flag indicating oscillatory behavior of fault flags in both schemes

ϕ_-^i = Penalty on false alarm for the MDP (Markov Decision Process)

ϕ_+^i = Penalty on missed detection for the MDP

φ = Penalty on conflict for the MDP

S = Set of states for the MDP $S = \{s_1, s_2, s_3, ..., s_N\}$

M = Set of actions for the MDP $M = \{\mu_+^1, \mu_-^1, \mu_+^2, \mu_-^2, NOOP\}$

$R(s_i)$ = Reward of state s_i in the MDP

$T(s_i, \mu_r^k, s_p)$ = Probability of transitioning from state s_i to s_p by executing action μ_r^k

I Introduction

Autonomous aerospace systems require increasingly sophisticated fault protection systems that maximize their ability to maintain a safe operational state in the presence of onboard system failures or environmental anomalies that pose risk or degrade performance. Several strategies have been proposed[1-5, 7] to detect, diagnose, and reconfigure in the presence of faults. The Markov Decision Process (MDP) and variants have been considered to manage discrete system models,[5, 8] while signal filtering, system identification, and adaptive control algorithms have been developed to manage physics-based (continuous) system models.[7] While many of the decisions made by discrete versus physics-based deliberation engines are distinct, many decisions or conclusions can also impact or overlap with the others.[12]

Considering the cost of space missions, their associated communication constraints, and the amount of risk involved due to hostile and uncertain deep space environment, it is desirable for space missions to have multiple fault detection schemes. In this situation, two or more detection schemes may occasionally render inconsistent decisions about the occurrence of a fault. Therefore, a conflict resolution algorithm is desirable.

This paper presents a formal language and protocol by which symbolic and physics-based fault management systems can share information to negotiate consistent decisions with respect to fault detection. Specifically, we present two methods of conflict resolution that minimize or eliminate discrepancies between the fault information obtained from two separate fault detection algorithms. Our methods apply to any pair of detection algorithms that satisfy corresponding assumptions. Our first method is based on initial threshold optimization with respect to a partial objective function and subsequent threshold updating that is optimal with respect to a specified cost function. While optimizing the thresholds, we make use of residual signals to minimize computational complexity. If the resulting minimum value of the objective function allows a persistent unresolved conflict, an error flag is generated that can be used to alert a human supervisor. The updating equations for thresholds attempt to keep the thresholds as close to the optimal values as possible without causing a conflict, with changes optimized within bounds imposed to achieve minimum acceptable performance criteria.

Our second conflict resolution method is based on a Markov Decision Process (MDP). This approach makes use of the reward function and discount factor to optimize changes in fault detection thresholds. This method does not require knowledge of residual signals but it is also more computationally-intensive.

In this paper we apply our conflict resolution strategy to a spacecraft example in which we model continuous time dynamics of the spacecraft and associated faults, as well as a limited number of discrete parameters (e.g., instrument on/off, valve open/shut). Our first fault detection scheme is based on an Interacting Multiple Model (IMM) framework that uses multiple models for the spacecraft to represent dynamics associated with certain specific fault conditions. With this strategy a bank of observers use the sensor data to compute the residuals for each fault models. The model with the lowest residual is assumed to be the true model and the fault condition that it relates to is considered to be the true condition of the spacecraft. Our second fault detection scheme is based on state transition system[8] with Markov assumption. Fault detection is based on the likelihood of reaching failure states given the transition probability table.

In the next section, we define the problem and present the two fault detection schemes in the context of a limited spacecraft fault detection model. In Section III, we present our threshold optimization and updating methods. Section IV shows simulation results, and Section V describes conclusions and future work.

II Basic Threshold Adjustment Approach to Conflict Resolution

To motivate the subsequent developments, we consider a system that uses two detection schemes to detect a particular fault. We assume that the performance of each scheme is represented by the probabilities of missed detection *P(MD)* and false alarm *P(FA)*, that are functions of adjustable parameters or thresholds associated with each scheme. Let $J(v^1, v^2)$ be a risk-based cost function which determines the combined performance of two fault detection schemes (1 and 2) as a function of two scalar fault parameters v^1 and v^2, one parameter for each scheme,

$$J(v^1, v^2) = a_-^1 P\left(FA^1 \mid v^1\right) + a_+^1 P\left(MD^1 \mid v^1\right) + a_-^2 P\left(FA^2 \mid v^2\right) + a_+^2 P\left(MD^2 \mid v^2\right) \qquad (1)$$

Here $a_-^1, a_+^1, a_-^2, a_+^2$ are positive weights that can be adjusted to emphasize missed detection and false alarms of either schemes. Note that $J(v^1, v^2)$ is a separable function of its arguments, i.e., it can be represented as $J(v^1, v^2) = J_1(v^1) + J_2(v^2)$.

Suppose now the two schemes make calls regarding the presence or absence of a particular fault, given the vector of current inputs and operating conditions, U. The fault flags of the two schemes are denoted by $b^1(v^1, U^1), b^2(v^2, U^2)$. The fault flag functions take binary values, either 0 or 1, depending on the inferred absence or presence of a fault.

The existence of a conflict corresponds to a situation in which $b^1(v^1, U^1) \neq b^2(v^2, U^2)$, i.e., one of the schemes indicates a fault and the other does not. To resolve an apparent conflict, a fault or no fault decision needs to be made. Such a call can be made by adjusting v^1 and v^2 so that $b^1(v^1, U^1) = b^2(v^2, U^2)$ and J is minimized.

$$\begin{aligned} & J(v^1, v^2) \rightarrow \min_{v^1, v^2} \\ & s.t. \\ & b^1(v^1, U^1) - b^2(v^2, U^2) = 0 \end{aligned} \qquad (2)$$

Due to the discontinuous nature of the fault flags, $b^1(v^1, U^1)$ and $b^2(v^2, U^2)$, which take binary values, it is not easy to solve the above optimization problem other than by a systematic grid search.

The case in which an easier solution strategy can be defined is when the parameter v^i is an additive threshold, i.e., the fault flags satisfy

$$b^i(V^i, U) = 0 \text{ iff } \mathrm{h^i}(U^i) - v^i \leq 0 \quad (\mathrm{i} = 1,2) \qquad (3)$$

Here, $h^i(U^i)$ are outputs (smooth functions) that we compare against the thresholds, and $h^i(U^i) - v^i$ are referred to as residuals. In case of multiple thresholds, where v^i and $h^i(U^i)$ are vectors, the inequalities in equation (3) are understood in component-wise sense. In this case, given that the objective function $J(v^1, v^2)$ is separable, the original optimization problem reduces to a finite number of smooth optimization problems that can be solved numerically. Modifications of these ideas will be used in the subsequent sections to define and illustrate two conflict resolution schemes.

A. Assumptions

We now discuss specific assumptions about the two fault detection schemes for which conflicts are to be resolved:

- A.1 Both detection schemes use numerical thresholds that determine the values of fault flags based on the information available and the inputs to the detection schemes.
- A.2 The residual signal which is the difference between an output and a threshold, based on which the fault flag is set, is known.
- A.3 The probabilities of false alarm (FA) and missed detection (MD) for each overlapping fault in both detection schemes are known (not necessarily analytically) and are monotonic functions of thresholds.

A.4 The communication of the information between the conflict resolution and fault detection schemes, and computations, are instantaneous.

These assumptions are justified for many practical detection schemes. A specific case study for spacecraft application is now introduced.

B. Spacecraft Case Study

In the paper we use a spacecraft case study to illustrate the use of two fault detection schemes and associated conflict resolution strategy. One of the fault detection schemes makes use of the physics-based dynamics model and the other makes use of a qualitative, logic-based model.

Consider a 1 DOF satellite attitude control system with two thrusters as shown in Figure 1. The dashed line shows the axis of rotation. Two thrusters are mounted in such a way that they produce equal and opposite forces resulting in torque about the axis of rotation.

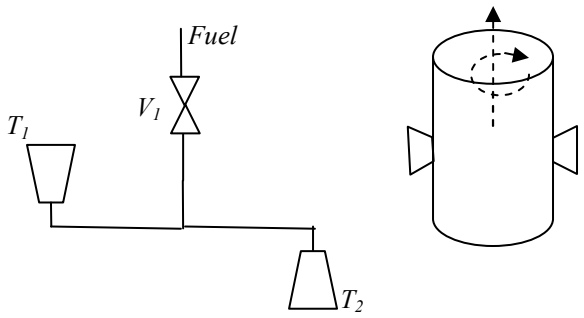

Figure 1: 1 DOF Satellite Schematic

The equations of motion for this system in normal (no-fault) mode can be written as

$$\begin{aligned} x(k+1) &= Ax(k) + Bu_c(k) + D\varepsilon(k) \\ y(k) &= Cx(k) + N\eta(k) \end{aligned} \tag{4}$$

Here, vector x represents orientation and angular velocity of the spacecraft, $u_c \in \{0, 1\}$ represents an impulsive thrust value of off (0) or on (1); y represents sensor readings, ε represents system disturbances, and η represents sensor noise. We assume disturbance and noise are normally distributed with zero mean values and known variances. Our physics-based fault detection technique relies on the Interacting Multiple Model (IMM) approach[9]. In this approach, we make use of the fact that, under certain faults (or combination of faults), the system has a specific and known dynamic model. Transitions between the set of possible dynamics models can be treated as discrete jumps. In this paper we consider only one fault case i.e. thrust failure. This leads to a discrete state *m(k)* taking values in state set $S = 0,1$. At each decision step k, transition probability $\pi_{ij}(k)$ of the model can be defined by

$$\begin{aligned} &\pi_{ij}(k) = P\{m(k) = j \mid m(k-1) = i\} \quad \forall i, j \in S \\ &\sum_{j \in S} \pi_{ij}(k) = 1 \quad i = 0,1 \end{aligned} \tag{5}$$

Now, consider a system model representing the fault states plus a nominal operation state (0).

$$\begin{aligned} x(k+1) &= A_j(k)x(k) + B_j(k)u_c(k) + D_j(k)\xi_j(k) \\ y(k) &= C_j(k)x(k) + N_j(k)\eta_j(k) \\ j &= 0,1 \end{aligned} \tag{6}$$

with the following values for disturbance and noise covariances

$$\begin{aligned} &\xi_j(k) = \mathbf{N}[\bar{\xi}_j(k), Q_j]; \eta_j(k) = \mathbf{N}[\bar{\eta}_j(k), R_j]; x_j(0) = \mathbf{N}[\bar{x}_0, \bar{P}_0] \\ &\bar{\xi}_j = \bar{\eta}_j = 0, Q_j = 0.01I_2, R_j = 0.05I_2, \bar{P}_0 = 10I_2 \quad \forall j \end{aligned} \tag{7}$$

Each cycle of IMM-based fault detection consists of four steps: mixing of estimates, model-conditional filtering, mode probability update and fault detection and diagnostics (FDD) logic, and combination of estimates.[9] Details of these steps for n-fault case are presented as Appendix. Fault flag b for the fault is set based on whether or not the likelihood of a mode has crossed corresponding threshold v[1].

$$b(k)=\begin{cases}1 & if: P\{m(k)=1\}\geq v^1\\ 0 & if: P\{m(k)=1\}<v^1\end{cases} \tag{8}$$

For our spacecraft, we define candidate probability functions for missed detection (MD) and false alarm (FA) for IMM. As mentioned earlier, we assume that these functions are monotonic with the value of v^1 (see Figure 2).

$$\begin{aligned}P(FA\mid v)&=\frac{1}{1+e^{10(v-0.4)}}\\ P(MD\mid v)&=\frac{1}{1+e^{-10(v-0.6)}}\end{aligned} \tag{9}$$

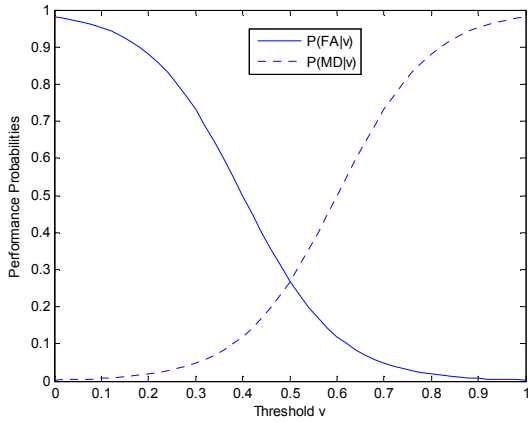

Figure 2: Example MD and FA Probabilities versus Threshold for IMM

Our second fault detection scheme for the same 1 DOF spacecraft is based on a qualitative model of the system that models components, their composition, and possible discrete value sets. As shown above in Figure 1, the main spacecraft components of interest are valves and thrusters. For fault detection purposes, we define the following logical clauses to describe the healthy system:

$$\begin{aligned}&(i)V_1\vee\neg T_1\vee\neg T_2\\ &(ii)T_1\wedge T_2\vee\neg V_1\\ &(iii)\neg T_1\vee T_2\\ &(iv)T_1\vee\neg T_2\end{aligned} \tag{10}$$

Here, V_1 represents that valve is *open* and its complement $\neg\ V_1$ represents a *closed* valve. Similarly T_i represents thruster *i* in *on* mode and its complement $\neg\ T_i$ represents an *off* thruster. In this model, we make use of the facts that the valve must be *open* for the thrusters to be *on* and the thrusters operate as a pair. This model can detect faults based on sensor readings. To identify faults, we may use the scheme presented by Williams et al.[8] We define the system by triplet $S=(\Pi,\Sigma,\Omega)$ for the valve V_1 where Π denotes the set of possible state features, Σ is the set of possible feature value sets, and Ω is a finite set of transitions between states. In our example, we have

$$\begin{aligned}&\Pi=\{status, cmdin, senout\}\\ &\Sigma=\{\{normal, failed\},\{open, close, none\},\{open, close, none\}\}\\ &\Omega=\{\tau_1,\tau_2,...,\tau_{(2\times3\times3)\times(2\times3\times3)}\}\end{aligned} \tag{11}$$

Each transition is characterized by transforming the state variables from one set of values to the same set or any other set reachable through a transition in Ω. Thruster states are a function of valve states; therefore we do not model thrusters with separate transitions. For each given state configuration, there is a set of possible transitions with associated probabilities, where the sum of all probabilities is equal to 1. This leads to the transition probability table of size *18×18* which we assume to be known.

If we represent O_t as the set of observations at time *t* and μ_t as the set of possible commands or actions, we can obtain the set of feasible states at time *t+1* as

$$S_{t+1} = \left(\bigcup_j \tau_j \left(S_t \bigcap S_{\mu t} \right) \right) \bigcap \Sigma \bigcap S_{Ot+1} \tag{12}$$

Once, S_{t+1} is computed, we can determine the most likely trajectories using Bayes rule

$$P(\tau \mid O_t) = \frac{P(O_t \mid \tau) P(\tau)}{P(O_t)} \tag{13}$$

In equation (13), if $\tau(S_{t-1})$ and O_t are disjoint sets then clearly $P(O_t| \tau) = 0$. Similarly, if $\tau (S_{t-1})$ is a proper subset of O_t then O_t is entailed and $P(O_t| \tau) = 1$, and hence the posterior probability of τ is proportional to the prior. If neither of the above two situations arises then $P(O_t| \tau) < 1$. Estimating this probability is intricate and requires more research.[8] Finally, the best estimate of current state is found using conflict directed best first search. Once the conclusion is made about the current state of the valve, fault flag *b* for the fault is set (1) if a valve has *status* = *failed* and is cleared (0) otherwise. Failure status is computed from observation *O* and hence the thresholds since the observed values depend upon comparison of sensor values against the thresholds

$$senout = \begin{cases} open & if : SensedVoltage > v^2 \\ close & if : SensedVoltage < v^2 \\ none & if : SensorFailed \end{cases} \tag{14}$$

This scheme can detect thruster failures based on valve failures. For example, let us consider the case where valve V_1 is used to supply fuel to thrusters. If *cmdin* = *open* but *senout* = *close* after an appropriate delay from initiation of the open instruction then the valve is failed in *close* mode; note that in this simplified model we assume a failed sensor will reliably give a *SensorFailed* status. The probabilities in Figure 3 depend upon the threshold, control command, and the previous state of the valve itself. For the conditions when probabilities change with threshold i.e. trying to open a closed valve or close an open valve, we can model the probabilities as functions of threshold in a similar way as we did for the IMM based detection scheme. In Figure 3 we present example relations between probabilities and threshold for the case when a closed valve is commanded to open (corresponding fault is stuck shut or *failed* in *close* mode).

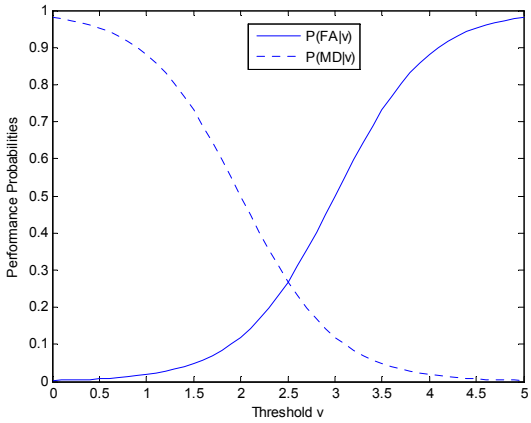

Figure 3: Performance Probabilities vs. Threshold for knowledge based detection Scheme

The above curves are based on 5 volt fuel pressure sensor where output is either 0 or 1 indicating the valve as *close* or *open*, respectively. Note that, as we increase the threshold, the probability of false alarm increases because there is greater chance of sensor reading 0 when it actually might be 1 and hence producing an incorrect detection of the valve as *close* when it actually is *open*. Analogous behavior is exhibited in probability of missed detection. Note that the chosen sigmoid functions are monotonic and given by

$$P(FA \mid v) = \frac{1}{1 + e^{-2(v-3)}}$$
$$P(MD \mid v) = \frac{1}{1 + e^{2(v-2)}} \tag{15}$$

Although the two fault detection schemes are based on different models, they share key properties that can be utilized. Specifically, we can manage their thresholds to resolve conflicts at the expense of decreased performance in

terms of MD and FA probabilities. Below we introduce a framework to recursively optimize thresholds so that the conflicts are minimized while maximizing performance in terms of MD and FA probabilities.

III Conflict Resolution

Consider the following conflict resolution cost function J_c.

$$J_c = \sum_i \left[a_-^i P\left(FA^i \mid v^i\right) + a_+^i P\left(MD^i \mid v^i\right)\right] + q\left|b^1\left(v^1, U^1\right) - b^2\left(v^2, U^2\right)\right| \qquad (16)$$
$$a_+^i > 0, a_-^i > 0, q > 0 \forall i : i \in \{1,2\}$$

In this equation, superscript i represents the detection scheme, a_+^i, a_-^i, q are penalty factors or weights, v^1 and v^2 are thresholds, and U^1 and U^2 are inputs to the detection schemes.

The objective function (16) has two main terms. The first term can be interpreted as a measure of risk incurred by changing the thresholds. The second term penalizes conflicts between the specific faults detected by the two schemes. Note that the second term depends on U^i which represents the command and sensor signals available to the detection scheme i.

A.1 Threshold Optimization

Define the first term of (16) as

$$J = \sum_i \left[a_-^i P\left(FA^i \mid v^i\right) + a_+^i P\left(MD^i \mid v^i\right)\right] \qquad (17)$$
$$i \in \{1,2\}$$

Note that

$$\frac{\partial J}{\partial v^i} = \frac{\partial J}{\partial P(FA^i)} \frac{\partial P(FA^i)}{\partial v^i} + \frac{\partial J}{\partial P(MD^i)} \frac{\partial P(MD^i)}{\partial v^i} \qquad (18)$$
$$i \in \{1,2\}$$

We assume nominal threshold values $\bar{v}^i$ are chosen to minimize J so that

$$v^i = \bar{v}^i \Rightarrow \frac{\partial J}{\partial v^i} = 0 \qquad (19)$$
$$\forall i \in \{1,2\}$$

We denote $J(\bar{v}^1, \bar{v}^2)$ by J^*. Note that J is a separable function i.e. we can optimize FA and MD probabilities for each fault detection scheme separately. The following result then follows.

Theorem 1: Based on the assumptions in section II-A, the minimum of J_c is achieved at thresholds v^1 and v^2 such that either $v^1 = \bar{v}^1$ or $v^2 = \bar{v}^2$ or both.

Proof: The proof follows from the observation that the fault flag changes if there is a change in the threshold exceeding the residual, monotonicity of $P(FA^i|v^i)$, $P(MD^i|v^i)$ as functions of v^i and positivity of the weights.

A.2 Residual-based Conflict Resolution

We now present our first method of conflict resolution using knowledge of residuals for threshold variation of two fault detection schemes detecting the same fault. Underneath is the mathematical formulation of threshold update equations for both fault detection schemes. With threshold updating, we are able to resolve fault decision conflicts in situations where the disagreement is not strong. By strong disagreement, we mean the cases for which the required change in thresholds causes the cost function increase beyond a given bound. We define the upper bound of cost function based the penalty weight on the conflict. For the cases where disagreement is strong, the thresholds are kept at their optimal values while an error signal is generated to alert a higher level supervisor that the conflict was not resolved.

The upper bound on cost function is defined as

$$J_{\max} = J^* + q \qquad (20)$$

The cost of resolving the conflict based on the knowledge of the residual signal for each scheme is given by

$$J^i = J(\bar{v}^i + \Delta v^i, \bar{v}^j)$$
$$i, j \in \{1,2\}, i \neq j \tag{21}$$

Here, Δv^i is the change in threshold for the i^{th} scheme required to resolve the conflict without changing the threshold for the other scheme.

The threshold for each scheme is updated based on the following equation

$$v^i(t+1) = \begin{cases} \bar{v}^i(t) + \Delta v^i & if: \min\left(J_{\max}, J^1, J^2\right) = J^i \\ & or: J^1 = J^2 < J_{\max}, i = 1 \\ \bar{v}^i(t) & otherwise \end{cases} \tag{22}$$
$$i \in \{1,2\}$$

B. Conflict Resolution based on the Markov Decision Process

In this subsection, we present a different solution to the conflict resolution problem based on Markov Decision Process (MDP) framework. In this case we eliminate assumption A.2 of Section II A which states that we have knowledge of residual signals. Hence we solve the problem of conflict resolution for any two schemes with independent fault detection such that each fault flag depends upon the value of a scalar parameter. Even though this parameter is not necessarily a threshold, we will refer to it as a threshold for consistency. The optimal MDP policy is then generated for adjusting the threshold(s) of the schemes to reach consensus between the fault flags.

The MDP framework uses a set of MDP states s_1, s_2, s_3… and leads to an optimal policy that can maximize the time-discounted utility of states i.e.

$$Policy^* = \arg\max_{\pi} E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid \pi\right] \tag{23}$$

The states of an MDP-based conflict resolution algorithm for a fault detected by two schemes can be defined as

$$S = \{s_1, s_2, s_3, ..., s_N\}$$
$$s_i = \{b_i^1, b_i^2, v_i^1, v_i^2\} \tag{24}$$
$$i \in \{1,2,..., N\}$$

Here, S contains states with all possible values of fault flags and thresholds related to the fault. We represent the threshold value set with a finite number of equally spaced discrete values v_i^k. Number of states N depends upon the size of Δv and the range of v. In particular, if number of possible values of each threshold is z, then $N = 4z^2$.

The actions are represented as

$$M = \{\mu_+^1, \mu_-^1, \mu_+^2, \mu_-^2, NOOP\} \tag{25}$$

and correspond to increasing (μ_+^1) or decreasing (μ_-^1) the threshold of first detection scheme by Δv^1; to increasing (μ_+^2) or decreasing (μ_-^2) the threshold of second detection scheme by Δv^2; and to no change (*NOOP*). Note that each action can result in four possible states. This is because we can change the threshold but cannot guarantee a specific transition in fault flags as the latter is determined by other signals the flags depend upon. Also NOOP results in the states with the same thresholds and same values of fault flags.

Rewards for each state can be represented as

$$R(s_i) = \exp\left(-\phi_+^1 P(MD^1 \mid v_i^1) - \phi_-^1 P(FA^1 \mid v_i^1) - \phi_+^2 P(MD^2 \mid v_i^2) - \phi_-^2 P(FA^2 \mid v_i^2) - \varphi\left|b_i^1 - b_i^2\right|\right)$$
$$i \in \{1,2,..., N\}, \phi_+^k, \phi_-^k, \varphi > 0 \tag{26}$$

Note that the reward depends upon MD and FA probabilities as well as conflicts in each state.

Using Monte Carlo simulations, one can construct the transition probabilities,

$$T = T(s_i, \mu_r^k, s_p)$$
$$k \in \{1,2\}, r \in \{+,-\}, i, p \in \{1,2,..., N\} \tag{27}$$

In the MDP, an optimal policy can be calculated using the value iteration algorithm. The policy that selects the optimal action may be found as

$$P^*(s_i) \in \arg\max_{k,r}\left(\sum_{p\neq i} T(s_i, \mu_r^k, s_p)\Gamma(s_p)\right) \tag{28}$$

There is a direct relationship between the utility of a state and the utilities of all the states that can be reached from that state in a single optimal action. This relationship can be expressed using the Bellman equation:

$$\Gamma_{t+1}(s_i) = R(s_i) + \max_{k,r}\left(\sum_{p\neq i} \gamma T(s_i, \mu_r^k, s_p)\Gamma_t(s_p)\right)$$
$$r \in \{+,-\}, k \in \{1,2\} \tag{29}$$

where $\Gamma_{t+1}(s_i)$ is the utility of state s_i at iteration $t+1$, $R(s_i)$ is the immediate reward of state s_i, and $T(s_i, \mu_r^k, s_p)$ is the probability of transitioning from state s_i to s_p by executing action μ_r^k. With this structure, the MDP computes the best available threshold setting (action) for each state. The computational complexity of value iteration algorithm is $5N^2$ per iteration of equation (29). The number of iterations required for convergence within a specified error tolerance depends upon the tolerance itself and the discount factor γ.

C. The Supervisor Alert

Since neither of our conflict resolution schemes guarantee 100% resolution of conflicts, it is important to have a supervisor (a human or software) that can handle strong conflicts. Detailed algorithm for such supervisor is beyond the scope of this paper however, one approach to generating an alert flag for a supervisor can be based on the following equation

$$e(t) = \prod_{l=t-k+1}^{t}\{c(l)\kappa(l)\}$$
$$c(t) = \left|b^i(v^i(t), U^i) - b^{\bar{i}}(v^{\bar{i}}(t), U^{\bar{i}})\right|, \kappa(t) = \left(c(t-1)c(t)\right)\left|b^i(v^i(t), U^i) - b^{\bar{i}}(v^{\bar{i}}(t-1), U^{\bar{i}})\right|$$
$$\bar{i}, i \in \{1,2\}, i \neq \bar{i} \tag{30}$$

Equation (30) has two interesting properties. First, it has a moving window that indicates a persistent conflict. The moving window is important to avoid intermittent anomalous situations that may be due to short term external or internal disturbances. The second feature is the use of an oscillation flag that avoids the generation of an error flag for the case of non-persistent fault flags causing a persistent conflict. The oscillation flag can also be used to detect particular failures such as power system failures causing fluctuations in voltages etc. The information about unresolved faults can be used to make adjustments in system models and/or in the fault detection schemes to account for a change in the environment or the system itself.

IV Simulation results

A. Residual-based Conflict Resolution

We tested our threshold adjustment conflict resolution strategies for the case study of section II. Recall that for the IMM based fault detection the residual is the difference between the threshold v^1 and the probability of fault mode (8) and for logic based fault detection; the residual is the difference between the threshold v^2 and the voltage output of the fuel pressure sensor (14). Based on our assumptions, the fault flag switches when the residual changes sign. For these simulations, we command the thruster *on/off* periodically with time period of 20 time steps and incorporate zero mean Gaussian noise in the residuals with variance of 60% of their maximum value i.e. variance of 3 for the 5 volt sensor output in logic based fault detection and variance of 0.6 for the probability of fault mode in IMM based fault detection. We did not inject the fault in our simulations. Therefore the conflict resolution is equivalent to mitigating a false alarm in one of the two schemes.

We selected the cost function as

$$J = 5P(MD^1 \mid v^1) + 7P(FA^1 \mid v^1) + 5P(MD^2 \mid v^2) + 7P(FA^2 \mid v^2) + \left|b^1(v^1, U^1) - b^2(v^2, U^2)\right| \tag{31}$$

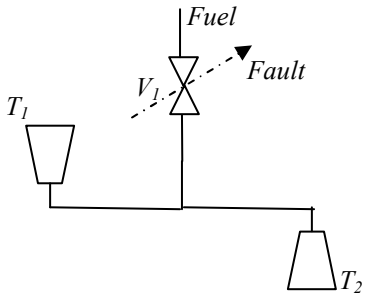

Figure 4: Thruster Valve System

Figure 5 illustrates the cost J defined in (17) based on our weights selected in (31). The value of J^* is 6.3129 and J_{max} = 7.3129 from (20) and (31). We measure the performance gain for our conflict resolution scheme as

$$PG = \frac{J_{\max} - J(v^1, v^2)}{J_{\max}} \times 100 \qquad (32)$$

Table 1 shows the simulation results for conflict resolution algorithm based on the knowledge of residuals. Note that the conflict is resolved 100% of the time and average performance gain is about 54%.

Table 1: Simulation Results for Residual Based Conflict Resolution

No. of Simulations	No. of conflicts incurred	No. of Conflicts Resolved	Average Performance Gain
50	5	5	54.15%
500	52	52	54.%
5000	554	554	54.05%
25,000	2463	2463	54.08%

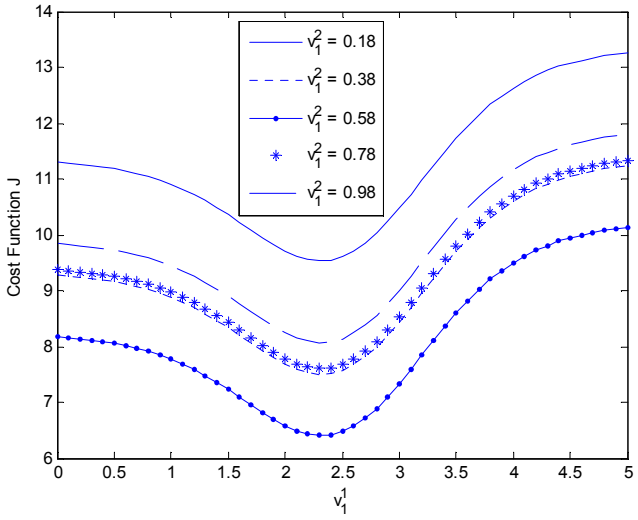

Figure 5: Cost as a Function of Thresholds

B. MDP based Conflict Resolution Method

We next consider the use of the MDP based conflict resolution algorithm. Also, in these simulations, the thruster is commanded periodically *on* and *off* with same time period as above. We used the following parameter values for our simulations.

$$\Delta v^1 = 0.25, \Delta v^2 = 0.05, \phi_+^1 = 5, \phi_-^1 = 7, \phi_+^2 = 5, \phi_-^2 = 7, \varphi = 100$$
$$z = 21, N = 1764, v^1 \in [0,5], v^2 \in [0,1], \gamma = 0.9, s_{initial} = \{0,0,2.25,0.55\}$$

Our numerical experiment consisted of executing the optimal policy on fault flags generated from both schemes based on similar residual signals as for simulation of residual based algorithm. The difference here though is that this algorithm does not make use of the residual signals and hence number of optimal actions required to resolve a given conflict is unknown.

Table 2: Simulation Results for MDP Based Conflict Resolution with 2 Actions per Conflict

No. of Simulations	No. of conflicts incurred	No. of Conflicts Resolved	Average Performance Gain
50	8	6	53.2%
500	41	34	53.1%
5000	413	312	52.97%
25,000	2215	1698	52.99%

Table 2 shows the simulation results where we allowed maximum of two actions per conflict. The performance gain is evaluated based on (32). Note that MDP is unable to resolve all the conflicts with in 2 changes in the thresholds. But if we allow more changes, more conflicts can be resolved as shown in Table 3 where we allowed MDP to execute up to 5 actions per conflict.

Table 3: Simulation Results for MDP Based Conflict Resolution with 5 Actions per Conflict

No. of Simulations	No. of conflicts incurred	No. of Conflicts Resolved	Average Performance Gain
50	3	3	54.04%
500	39	39	54.04%
5000	386	386	54.04%
25,000	2052	2052	54.04%

From the comparison of results in tables 2 and 3, it seems as if we allow sufficient number of actions for an MDP based resolution scheme, we can get 100% conflict resolution. It is important to understand, however, that this is not always the case. For example, when the conflict is strong, the required change in thresholds will be so large that the reward obtained for resolving the conflict will be less than loss of reward due to sub-optimal threshold.

V Conclusions

We have presented two conflict resolution algorithms capable of resolving differences in faults detected by diverse fault detection schemes; we presented two candidate fault detection schemes for our study, one based on IMM and one based on Markovian state transition system. Both our conflict resolution schemes share a common resolution strategy: they change missed detection vs. false alarm thresholds for one or both fault detection algorithms as a method of converging on a common fault set. In the first conflict resolution method, we optimized the changes in thresholds with respect to a cost function that takes into account not only the conflicts but also the probabilities of missed detection and false alarms for both schemes and that uses residuals to update threshold values. In our second strategy, we optimize the change in thresholds using a Markov Decision Process based on rewards, transition probabilities of fault flags, and a discount factor but without knowledge of residuals. We have demonstrated the ability of both conflict resolution algorithms to resolve conflicts using a simple example of spacecraft thruster failure. Simulation results show that our approaches are able to resolve conflicts, with the residuals method offering a faster solution and the MDP method offering a more general resolution method not dependent on knowledge of residuals. In future work we plan to extend our models to accommodate more than two fault detection schemes and to handle multiple faults, initially with independent thresholds then ultimately with interdependencies.

Appendix: The IMM Cycle[9]

The four steps for IMM-based fault detection based on the system given in Equations (6) and (7), and generalized to the case of n faults are given below (n = 1 in the case study which we considered). These steps include mixing of estimates, model-conditional filtering, mode probability update and FDD, then combination of estimates.

1. Interaction/Mixing of the estimates (for $j = 0,1,2,\ldots,n$):

   predicted mode probability: $\mu_j(k+1|k) \equiv P\{m_j(k+1) \mid y^k\} = \sum_i \pi_{ij}\mu_i(k)$

   mixing probability: $\mu_{i|j}(k) \equiv P\{m_i(k) \mid m_j(k+1), y^k\} = \pi_{ij}\mu_i(k) / \mu_j(k+1|k)$

   mixing estimate: $\hat{x}_j^0(k|k) \equiv E[x(k) \mid m_j(k+1), y^k] = \sum_i \hat{x}_i(k|k)\mu_{i|j}(k)$

   mixing covariance:
   $$P_j^0(k|k) \equiv \mathrm{cov}[\hat{x}_j^0(k|k) \mid m_j(k+1), y^k]$$
   $$= \sum_i \left[P_i(k|k) + \left[\hat{x}_j^0(k|k) - \hat{x}_i(k|k)\right]\left[\hat{x}_j^0(k|k) - \hat{x}_i(k|k)\right]^T \right]\mu_{i|j}(k)$$

2. Model-conditional filtering (for $j = 0,1,2,\ldots,n$):

   projected state (from k to $k+1$):

   $\hat{x}_j(k+1|k) \equiv E[x(k+1) \mid m_j(k+1), y^k] = A_j(k)\hat{x}_j^0(k|k) + B_j(k)u_c(k) + D_j(k)\overline{\xi}_j(k)$

   projected covariance:

   $P_j(k+1|k) \equiv \mathrm{cov}[\hat{x}_j(k+1|k) \mid m_j(k+1), y^k] = A_j(k)P_j^0(k|k)A_j(k)^T + D_j(k)Q_jD_j(k)^T$

   measurement residual:
   $$\nu_j \equiv y(k+1) - E[y(k+1) \mid m_j(k+1), y^k]$$
   $$= y(k+1) - C_j(k+1)\hat{x}_j(k+1|k) - \overline{\eta}_j(k+1)$$

   residual covariance: $S_j \equiv E[\nu_j \mid m_j(k+1), y^k] = C_j(k+1)P_j(k+1|k)C_j(k+1)^T + R_j$

   filter gain: $K_j = P_j(k+1|k)C_j(k+1)^T S_j(k+1)^{-1}$

   updated state: $\hat{x}_j(k+1|k+1) \equiv E[x(k+1) \mid m_j(k+1), y^{k+1}] = \hat{x}_j(k+1|k) + K_j\nu_j$

   updated covariance:

   $P_j(k+1|k+1) \equiv \mathrm{cov}[\hat{x}_j(k+1|k+1) \mid m_j(k+1), y^{k+1}] = P_j(k+1|k) - K_j(k+1)S_j(k+1)K_j(k+1)^T$

3. Mode probability update and FDD logic (for $j = 0,1,2,\ldots,n$):

   likelihood function: $L_j(k+1) = \dfrac{1}{\sqrt{|(2\pi)S_j(k+1)|}} \exp[-\dfrac{1}{2}\nu_j(k+1)^T S_j^{-1}(k+1)\nu_j(k+1)]$

   mode probability: $\mu_j(k+1) \equiv P\{m_j(k+1) \mid y^{k+1}\} = \dfrac{\mu_j(k+1|k)L_j(k+1)}{\sum_i \mu_i(k+1|k)L_i(k+1)}$

   FDD decision: $\mu_j(k+1) = \max_i \mu_i(k+1) \begin{cases} > threshold & fault_j \\ \le threshold & no\_fault \end{cases}$

4. Combination of estimates:

   aggregated estimate: $\hat{x}(k+1|k+1) \equiv E[x(k+1) \mid y^{k+1}] = \sum_j \hat{x}_j(k+1|k+1)\mu_j(k+1)$

   aggregated covariance:

   $$P(k+1|k+1) \equiv E\left[\left[x(k+1) - \hat{x}(k+1|k+1)\right]\left[x(k+1) - \hat{x}(k+1|k+1)\right]^T \mid y^{k+1}\right]$$
   $$= \sum_j \left[P_j(k+1|k+1) + \left[\hat{x}(k+1|k+1) - \hat{x}_j(k+1|k+1)\right]\left[\hat{x}(k+1|k+1) - \hat{x}_j(k+1|k+1)\right]^T\right]\mu_j(k+1)$$

Acknowledgement

The authors would like to thank the Institute of International Education for their fellowship and travel support for the first author.

References

[1]Meier, L., Ross, D.W. and Glaser, M.B., "Evaluation of the Feasibility of Using Internal Redundancy to Detect and Isolate Onboard Control Data Instrumentation Failures," Tech. Report AFFDL-TR-70172, Wright-Patterson Air Force Base, Dayton, Ohio, Jan., 1971.

[2]Deyst, J.J. and Deckert, J.C., "Maximum Likelihood Failure Detection Techniques Applied to the Shuttle RCS Jets," *J. Spacecraft and Rockets,* AIAA, Vol. 13, 65-74, 1976.

[3]Clark, R.N., Fosth, D.C. and Walton, W.M., "Detecting Instrument Malfunctions in Control Systems," *IEEE Trans. Aerospace and Electronic Systems,* IEEE, Vol. AES-11, 465-473, 1975.

[4]Walker, B.K. and Gai, E., "Fault Detection Threshold Determination Technique Using Markov Theory," *Journal of Guidance, Control and Dynamics,* AIAA, Vol. 2, 313-319, July-Aug. 1979.

[5]Patton, R.J., Frank, P.M., and Clark, R.N., *Fault Diagnosis in Dynamic Systems: Theory and Applications,* Prentice-Hall, Englewood Cliffs, NJ, 1989.

[6]Russell, S. and Norvig, P., *Artificial Intelligence: A Modern Approach,* 2nd Edition, Prentice-Hall, Upper Saddle River, New Jersey 07458, 2005.

[7]Zhang, Y., Jiang, J., "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual Reviews in Control*, Elsevier Ltd, Volume 32, Issue 2, December 2008, Pages 229-252.

[8]Williams, C**.**B**.,** and Nayak, P.P., "A Model-Based Approach to Reactive Self-Configuring Systems," in *Proceedings of AAAI-96,* pages 971-978, AAAI, AAAI Press, Cambridge, Mass., 1996.

[9]Zhang, Y.M. and Jiang, J, "Integrated Active Fault Tolerant Control Using IMM Aproach". *IEEE Transactions on Aerospace and Electronic Systems,* IEEE, Vol. 37, No. 4, 1221-1235, October 2001.

[10]Frank, P.M. and Ding, X., "Survey of Robust Residual Generating and Evaluation Methods in Observer-Based Fault Detection Systems," *Journal of Process Control,* Elsevier Ltd, Vol. 37, No. 6, 403-424, 1997.

[11]Patton R. J., Lopez-Toribio C. J., & Uppal F. J., "Artificial Intelligence Approaches to Fault Diagnosis," *Applied Mathematics and Computer Science*, Technical University of Zielona Gora, Poland, Vol. 9, No. 3, 471-518, 1999.

[12]Nasir, A. and Atkins, E.M. "Fault tolerance for Spacecraft Attitude Management," AIAA *Guidance, Navigation, and Control Conference,* Toronto, Ontario, Aug. 2-5, 2010 (AIAA-2010-8301).