A Hybrid System Approach towards Redundant Fault-Tolerant Control Systems

Zhenyu Yang
Department of Control Engineering
Aalborg University
Fredrik Bajers Vej 7C, 9220 Aalborg, Denmark
E-mail: yang@control.auc.dk

Abstract

This paper discusses the verification problem of Redundancy Management Systems (RMS) in fault-tolerant control by using a hybrid system approach - the Discrete-Event-System (DES) abstracting strategy. The qualitative fault-tolerant criteria can be formally verified if a DES model is abstracted from the continuous/discrete-time dynamical system in a consistent way. The acquisition of the DES model and verification of fault-tolerant criteria are illustrated based on a concrete RMS of a redundant flight control system.

1 Introduction

Fault tolerance in control is the ability of control systems to cope with faults in components of controlled plants and ensure that faults do not develop into failures or emergencies [4, 9]. In order to carry out the fault tolerance, the *redundancy techniques* are utilized popularly in Fault-Tolerant Control Systems (FTCS), such as *direct redundancy* and *analytical (indirect) redundancy*. Consequently, some specific systems referred to as *Redundancy Management Systems (RMS)* are employed in order to monitor and manage these redundant signals/components. For example, one redundant control loop of a triplex-channel Fly-By-Wire (FBW) flight control system is illustrated in Fig.1. These modules called *Signal Monitor and Redundancy Management (SMRM)* monitor the redundant input signals and then output a consolidated signal to the following components. These SMRM modules carry out the function of RMS, and can be implemented by logic circuits or software programs.

Once we just consider the system components as digital controller box, sensor box, actuator box and dynamical system as shown in Fig.1, then the redundant control system fits into the conventional control system scheme well. However, the analysis and synthesis of redundant fault-tolerant control systems not only need to consider the conventional control problem, but also need to consider the RMS problem as well. It is obvious as shown in Fig.1 that the correct operation of SMRM modules plays a prominent role in the whole system stability and performance, even though we could have a strong robust digital controller. However, little attention has been payed to the systematic analysis and synthesis problems related to RMS in fault-tolerant control research area, although numerous work can be found in some traditional topics such as model-based FDI and reconfigurable control [4, 9]. Some fundamental problems about the RMS, such as

- The systematic and efficient way to design a RMS;
- Verification of the correctness of a designed RMS;
- The reliability of a designed RMS, and
- The cooperation between reconfigurable digital controller and RMS,

are quite far away from being well solved. The analysis and synthesis of practical RMS heavily depend on simulations and experimental tests regarding to the empirical strategy. Moreover, many embedded RMS are required to satisfy some fault-tolerant criteria in system level, such as the requirement: $FO/FO/FS$ for the quadruple-channel RMS, where $FO/FO/FS$ means the system should operate normally when one or two faults occurred inside the system, and the system should operate in a safe mode when there are more than three faults occurred. As we know, there is a little systematic work to say that a designed RMS makes the requirement ($FO/FO/FS$) satisfied in the dynamical system level, although there are many work on data RMS from the algorithm's point of view in the research area of computer science [1, 5].

With respect to the system's structure as shown in Fig.1, the RMS is an embedded real time system and its operation includes qualifizing fault events from the continuous/digital signals, determining the selection of valid signals regarding to the detected fault information, and deciding the voting rules so as to output a consolidated signal. The whole closed-loop system actually is a typical hybrid control system [2, 6, 8, 11]. Therefore, in this paper we discuss the verification

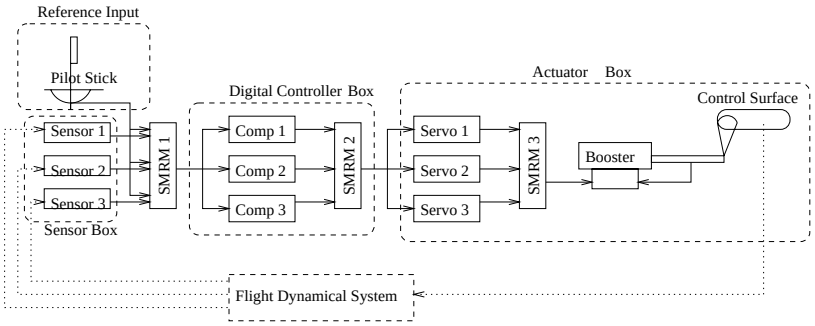

Figure 1: One Loop of a Triplex FBW Control System

problem of RMS by using a hybrid system method - the Discrete-Event-System (DES) abstracting approach [3, 6, 7, 8, 11], i.e., once an DES model of the closed-loop dynamical system can be abstracted, the verification of qualitative fault-tolerant criteria is equivalent to proving an invariant predicate in this DES model.

The rest of this paper is organized as: Section 2 introduces a quadruple-channel RMS which has been used in a practical flight control system. Section 3 reviews briefly the I/O automaton based abstracting approach for hybrid systems [11]; Section 4 discusses the acquisition of an I/O automaton abstraction of the concrete RMS; Section 5 discusses the verification of the considered RMS based on the I/O automaton abstraction; And at last section 6 is conclusions.

2 A Flight Redundancy Management System

A RMS consists of three functional parts: Monitoring Algorithm (MA), Voting Algorithm (VA) and RMS Logic. The MA monitors redundant signals and generates residuals according to some specific numerical algorithms. The VA generates a voted signal by consolidating proper signals according to some numerical algorithms. While the RMS Logic plays the "brain" role in the RMS, it carries out the Fault Detection and Isolation (FDI) by some logical rules based on the residuals generated from MA, and determine the usage of different voting rules with respect to the detected fault information. The RMS Logic should keep the whole RMS operating in a correct and consistent way.

Here we consider a concrete quadruple-channel RMS which has been used in a practical FBW flight control system. Denote signal-values from the four redundant channels at one sampling instant as x_1, x_2, x_3, x_4 respectively, and assume there is $x_1 \geq x_2 \geq x_3 \geq x_4$ [1].

The MA is to calculate the absolute difference between x_i and x_{i+1} for $i = 1, 2, 3$, i.e., three residuals δ_i will be generated by

$$\delta_i \hat{=} |x_i - x_{i+1}|, \quad i = 1, 2, 3. \tag{1}$$

The VA has the following rules:

- V0: Output the priorly assigned Failure-Safe-Value;
- V1: Output the value of available signal;
- V2: Output the mid-value of available signals;
- V3: Output the average of available signals;
- V4: Output the average of two mid-value signals.

The RMS Logic for FDI has the rules:

- RM1: If there are 4 valid signals initially and $\bigwedge_{i=1}^{3}(\delta_i < T_a)$, then declare all signals valid;
- RM2: If there are 4 valid signals initially and only δ_1 or δ_3 greater than the predefined threshold T_a, i.e., 1 vs. 3 cases[2], then declare the one signal invalid and the other three valid;
- RM3: If there are 3 or 2 valid signals initially and no δ_i (i =1 or 2) greater than T_a, then declare all valid;
- For cases except those suitable directly for RM1 and RM2, the Multiple Failure Logic (MFL) is needed.

In order to deal with possible multiple failure cases, each redundant channel has equipped a hardware/software In-line-Monitor (ILM) component, and which monitors the status of current channel. The ILM information (valid or invalid signals) participate in MFL as:

- RM4: All signals with valid ILM are declared valid, while the others are condemned;
- RM5: If the signals with valid ILM are split as 2 vs. 2, or 1 vs. 1 vs. 1 vs. 1, or 1 vs. 1 vs. 1, or 1 vs. 1 cases, condemn all signals;
- RM6: If the signals with valid ILM are split 2 vs. 1 case, condemn the 1;

[1] Here we consider the analog signals, the proposed method can also deal with digital signals.

[2] If the residual of two adjoining signals is greater than the threshold, we say there is a 1 vs. 1 case.

The RMS Logic for signal-voting has rules:

- Isolate all invalid signals, and let only valid signals into voting algorithm;
- RV1: If there is no valid signal, select V0;
- RV2: If there is only one valid signal, select V1;
- RV3: If there is 2 valid signals, select V2;
- RV4: If there is 3 valid signals, select V3;
- RV5: If there is 4 valid signals, select V4;

Here we assume that once a signal is declared invalid, then the corresponding channel will be invalid. An RMS is said *correct* when (1) it detects and isolates faulty signals/channels correctly; and (2) it makes the fault-tolerant criterion satisfied within any possible operating situations.

It should be noted that the invalid signals declared by RMS don't mean all these signals are really faulty, because the real value of considered signal is usually unknown in practice. In the following, the relationship between (real-life) faulty and (algorithm-declared) invalid signals is explored qualitatively by using the I/O automaton based approach for hybrid systems [11].

3 The I/O Automaton Based Approach for Hybrid Control Systems

According to the strategy of using the DES abstraction in analysis and synthesis of hybrid control systems [6, 7, 8, 11], once an DES model can be abstracted for the continuous-time dynamical plant and A-D interface, the analysis and synthesis of the logic part can be carried out by the DES theory [3, 10]. The DES abstraction acts as the interface for the cooperative analysis and design within two different domains. In order to describe the DES abstraction which maybe includes the abstraction of a set of continuous/discrete-time controllers, a kind of I/O automata, named DES I/O automata was proposed in [11]:

Definition 1 [11]: A DES I/O automaton (DE-IOA) is defined as a tuple $M \hat{=} \langle Q, U, \Sigma_{int} \cup \Sigma_{ext}, \varphi, Q_0, U_0, \lambda, F \rangle$, where Q is a finite set of *discrete modes*, each $q \in Q$ represents a kind of continuous-time dynamic status of the considered system. U is a finite set of *control symbols*, each $u \in U$ represents a kind of continuous/discrete-time control status of considered system. $(q, u) \in Q \times U$ is referred to as a state of the DE-IOA. Σ_{int} is an *internal event set*, and Σ_{ext} is an *external event set*. If the operating continuous/discrete-time controller switches at one moment, then we say there occurs an external event at this moment. If the system's state trajectory runs from one discrete mode into another, then we say there occurs an internal event at the crossing moment. $\varphi \hat{=} \varphi_{int} \vee \varphi_{ext}$ is a partial *state transfer function*, where

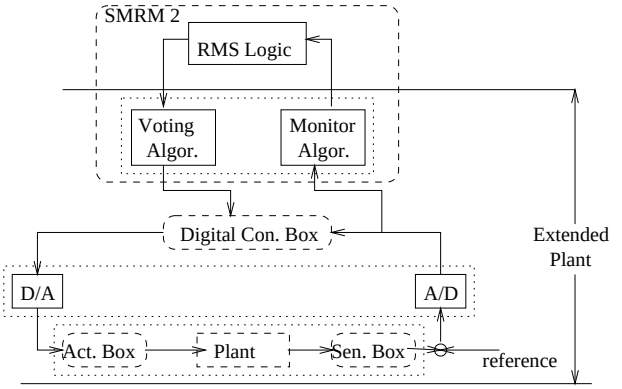

Figure 2: The Hybrid Framework of an Embedded RMS

$\varphi_{int} : (Q \times U) \times \Sigma_{int} \mapsto Q \times U$ is an internal state transfer function like[3] $\varphi_{int}((q, u), en_{qq'}) = (q', u)$, for $en_{qq'} \in \Sigma_{int}$, and $\varphi_{ext} : (Q \times U) \times \Sigma_{ext} \mapsto Q \times U$ is an external state transfer function. Q_0 and U_0 are the sets of initial discrete modes and controls symbols, respectively. $\lambda : Q \times Q \mapsto \Sigma_{int}$ is the *output function*, and $F \subseteq Q \times U$ is a *marked set*, representing the expected status of the whole system.

The *language* and *marked language* generated by DE-IOA M are defined as

$$L(M) \hat{=} \{s \mid \varphi((q_0, u_0), s) \text{is defined}, s \in \Sigma^\star, q_0 \in Q_0, u_0 \in U_0\}, \tag{2}$$

$$L_F(M) \hat{=} \{s \mid \varphi((q_0, u_0), s) \text{is defined and } \varphi((q_0, u_0), s) \in F, s \in \Sigma^\star, q_0 \in Q_0, u_0 \in U_0\}. \tag{3}$$

According to the operating principle of DE-IOA [11], the symbols of external events and internal events are mutually crossed in languages $L(M)$ and $L_F(M)$. Let $del(D)(L)$ denote the *projected language* obtained from strings in L by removing all symbols belong to D [7], then the input and output languages of M, denoted as $L_{in}(M)$ and $L_{out}(M)$ respectively, can be expressed as

$$L_{in}(M) \hat{=} del(\Sigma_{int})(L(M)),\ L_{out}(M) \hat{=} del(\Sigma_{ext})(L(M)). \tag{4}$$

It is obvious that $L_{in}(M) \subseteq \Sigma_{ext}^\star$ and $L_{out}(M) \subseteq \Sigma_{int}^\star$.

4 DE-IOA Abstraction of the Embedded RMS

The embedded RMS implemented by the SMRM 2 module in practical system fits into the framework of hierarchical hybrid control systems as shown in Fig.2. Assume these quadruple channels are already clock synchronized, and the DE-IOA model is denoted as $M_{rms} \hat{=} \langle Q, U, \Sigma_{int} \cup \Sigma_{ext}, \delta, Q_0, U_0, \lambda, F \rangle$. In the following we discuss the acquisition of this DES model.

[3] Here we only consider the deterministic systems.

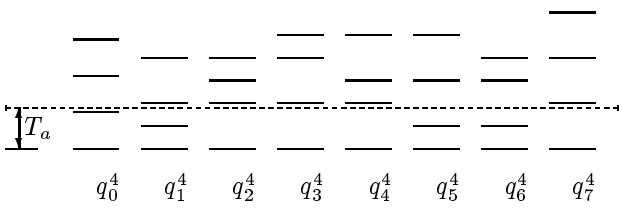

Figure 3: Different Situations under Four Valid Channels

4.1 Control Alphabet and External Events

With respect to the voting algorithms, let symbols V_0, V_1, V_2, V_3, V_4, represent the five different voting rules V0, V1,V2,V3,V4 respectively, and these symbols consist of the control alphabet U of the desired DE-IOA model. The decision events generated by the RMS logic represent two synchronized activities: (1) isolating invalid signals; and (2) selecting proper voting rule from V0, V1,V2,V3 and V4. If we denote decision events as D_i^j for $i, j = 1, 2, 3, 4$ and $i \geq j$, they also can be expressed as two prioritized synchronized events, denoted as $C_i^{i-j} \| (V_i \rightarrow V_j)$, where event C_i^{i-j} represents the activity of isolating $i-j$ (invalid) signals from current i signals, i.e., there is(are) only j signal(s) will enter the voting algorithm. The event $V_i \rightarrow V_j$ represents the switch of voting rules from V_i to V_j. Once the decision events can be accepted and executed correctly by the open DE-IOA model, then these decision events equal the external events in the closed DE-IOA model. Therefore the external event set can be obtained as $\Sigma_{ext} \hat{=} \{D_1^1, D_2^1, D_2^2, D_3^1, D_3^2, D_3^3, D_4^1, D_4^2, D_4^3, D_4^4\}$, and these analysis can also be employed to define the function φ_{ext} in M_{rms}.

4.2 Discrete Modes and Internal Events

We assume there are four valid channels at the beginning, then at one sampling time instant, there are eight possible different situations with respect to the monitoring algorithm (1). These different situations are labeled by discrete modes as illustrated in Fig.3 respectively. Since the initial state (Q_0, U_0) is (q_0^4, V_4), by following the RM2 two different internal events can be defined as: the event e_1^4 denotes the mode transition $q_0^4 \rightarrow q_1^4$, and the numerical criterion for determination of the occurrence of this event is $(\delta_{12} \geq T_a) \wedge (\bigwedge_{i=2}^{3} \delta_{i,i+1} < T_a)$; the event e_2^4 denotes the mode transition $q_0^4 \rightarrow q_2^4$, and numerical criterion for this event is $(\delta_{34} \geq T_a) \wedge (\bigwedge_{i=1}^{2} \delta_{i,i+1} < T_a)$. For the other situations, RM4 and RM5 are needed for further exploration. For example, within the situation of 1 vs. 1 vs. 2, such as the mode q_3^4, more discrete modes need to be refined according to the ILM information, e.g., the refined modes corresponding to the mode q_3^4 are defined as shown in Fig.4, where $\circ$ denotes the invalid ILM signal. Within this case, some internal events can be defined as e_{3i}^4, which represents the transition $q_0^4 \rightarrow q_{3i}^4$ for

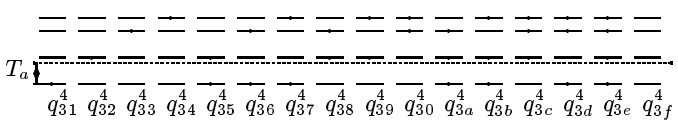

Figure 4: Refined Modes Corresponding to q_3^4 with Respect to ILM Information

$i = 1, \cdots, f$. With respect to the RMS logic, the corresponding synchronized external events for these defined internal events are:

External Event	Corresponding Internal Event
D_4^2	$e_{31}^4, e_{32}^4, e_{35}^4, e_{3f}^4$
D_4^3	$e_{3a}^4, e_{3b}^4, e_{3c}^4, e_{3d}^4$
D_4^4	all those not considered by D_4^2, D_4^3

Then we get a refined sub-DE-IOA model which is related to the mode q_3^4 as illustrated in Fig.5, where q_1^2, Q_1 and Q_0 represent the situations that there are 2, 1 and 0 valid channels inside the considered system respectively. The internal event set of this sub-DE-IOA is $\Sigma_{int}^{q_3^4} \hat{=} \{e_{3i}^4\}_{i=1}^f$ and external event set is $\Sigma_{ext}^{q_3^4} \hat{=} \{D_4^2, D_4^3, D_4^4\}$.

4.3 DE-IOA Model of the Closed-Loop System

The DE-IOA model for the whole closed-loop system can be obtained by combining all sub-DE-IOA models obtained though above analysis for all possible situations. The marked set of the obtained model is $F \hat{=} \{(q_0^4, V_4), (q_1^3, V_3), (q_1^2, V_2), (q_1^1, V_1), (Q_0, V_0)\}$. If we abstract this DE-IOA model in a high abstraction level, i.e., in the forms of (Q_i, V_k), where Q_i for $i = 0, 1, 2, 3, 4$ represents the case that there is(are) i valid channel(s), we can get a abstracted DE-IOA model as shown in Fig.6. Because this DE-IOA model represents the operation of the closed-loop system, some states denoted as dash-boxes in Fig.6 do not appear in the closed-loop DE-IOA operation.

Remark 2: The detailed analysis in the model set-up is necessary in exploring different FDI mechanisms for different situations, because this exploration is very important in (1) getting the fault information from (declared) invalid signals; and (2) exploring the probabilistic reliability of each action in order to explore the whole system reliability.

5 Verification of the Considered RMS

The qualitative fault-tolerant criterion for the considered RMS is the requirement: $FO/FO/FS$, which means: (1) no matter what kind of situations, the system should operate in the nominal or safe status; (2) the system should accommodate at least two sequential/parallel faults in the nominal operation. In following, we assume: (1) the majority-decision strategy and

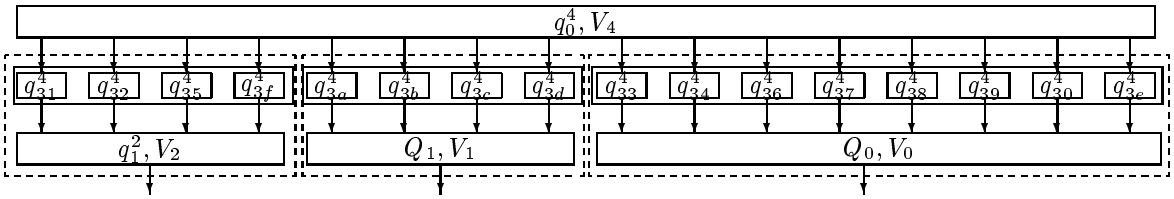

Figure 5: Sub-DE-IOA Model Related to q_3^4 (here q_{3i}^4 abbreviates (q_{3i}^4, V_4))

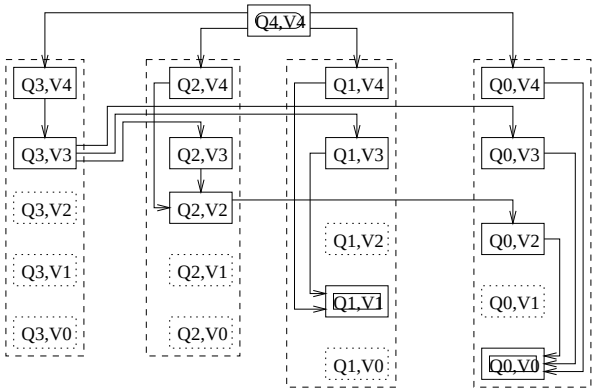

Figure 6: The DE-IOA Model of the Closed-Loop System in (Q_i, V_j) Level

ILM information in channel FDI are both correct; (2) once two adjoining signals are healthy, their absolute distance is no more than the threshold T_a; and (3) if there is only one channel left, the signal in this channel will be always correct.

5.1 The DE-IOA Model in Fault Event Level

It is obvious that the declaration of invalid channels by RMS does not mean all these invalid channels are really faulty. Some internal events defined in above have explicit fault information with respect to the majority-decision and ILM information, such as the event e_1^4 (representing the 1 vs. 3 case), while some other internal events have no explicit fault information, such as the event e_7^4 (representing the 1 vs. 1 vs. 1 vs.1 case), which just means that there are possible 3 or 4 faults. With respect to the concrete RMS introduced in section 2, relationships between internal events in DE-IOA model and fault events are expressed in table 1.

Let a fault event be denoted as E, two sequential fault events as $E.E$, and two parallel fault events as $E||E$. Furthermore, let E^k represents there are k fault events, no matter they are sequential or parallel, then we have

Rule 1: $E.E \Rightarrow E^{1+1} \hat{=} E^2,\ E||E \Rightarrow E^{1+1} \hat{=} E^2,$

With respect to table 1, for $i, j \in \{1, 2, 3, 4\}$, we have

Rule 2 : $E_j^i \Rightarrow E^i,\ \ ?E_j^j \Rightarrow E^{j-1} \vee E^j,$
$??E_4^4 \Rightarrow E^4 \vee E^3 \vee E^2.$

5.2 Verification of the Considered RMS

In order to set up the connection of states with the language of the DE-IOA model, we use the notation $(q_{jk}^i, V_i) \wedge \alpha \Rightarrow \varphi((q_{jk}^i, V_i), \alpha)$, for $q_{jk}^i \in Q$ and $\alpha \in L(M_{rms})$ to denote that the DE-IOA model can reach the final state $\delta((q_{jk}^i, V_i), \alpha)$ from the initial state (q_{jk}^i, V_i) by following the string α which belongs to the language of M_{rms}, and it has the property:

Rule 3: $(q_{jk}^i, V_i) \wedge s.t \Rightarrow ((q_{jk}^i, V_i) \wedge s) \wedge t,$
$\forall s.t \in L(M_{rms}),$
$(q_{jk}^i, V_i) \wedge (s \vee t) \Rightarrow ((q_{jk}^i, V_i) \wedge s)$
$\vee ((q_{jk}^i, V_i) \wedge t),\ \forall s, t \in L(M_{rms}).$

Due to the criterion of $FO/FO/FS$, any state of the built M_{rms} can be abstracted as one of three states: *nominal_state*, *safe_state* and *faulty_state*, therefore, we have the following:

Rule 4 : $(q_0^4, V_4) \Rightarrow$ *nominal_state*,
$(\forall \alpha \in L(M_{rms}), (q_0^4, V_4) \wedge \alpha \Rightarrow ((q_1^i, V_i)$
$\wedge (1 \leq i \leq 3))) \Rightarrow$ *nominal_state*,
$(\forall \alpha \in L(M_{rms}), (q_0^4, V_4) \wedge \alpha \Rightarrow (Q_0, V_0))$
$\Rightarrow$ *safe_state*,
$(\forall \alpha \in L(M_{rms}), (q_0^4, V_4) \wedge \alpha \Rightarrow ((Q_i, V_j)$
$\wedge (i \neq j))) \Rightarrow$ *faulty_state*,

Define a counter operator $\Im : L_{out}(M_{rms}) \mapsto \{E^k\}_{k=0}^N$, where $L_{out}(M_{rms})$ is the output language of M_{rms} as defined in (4). $\forall \alpha \in L(M_{rms})$, the result of $\Im(del(\Sigma_{ext})(\alpha)) = E^k$ for $k = 1, 2, 3, 4$ means that the string α indicates that there is(are) k fault event(s) implicitly. If we assume the operator $\Im$ has the property:

Rule 5 : $\Im(s.t) = \Im(s).\Im(t),\ \forall s.t \in L_{out}(M_{rms}),$
$\Im(u \vee w) = \Im(u) \vee \Im(w),\ \forall u, w \in L_{out}(M_{rms})$

then the verification of the criterion $FO/FO/FS$ is equivalent to proving that the predicate

$$\begin{aligned} &\forall \alpha \in L(M_{rms}),\ ((q_0^4, V_4) \wedge \alpha \wedge (\Im(del(\Sigma_{ext})(\alpha)) \\ &= E^k \wedge 0 \leq k \leq 2) \Rightarrow \textit{nominal\_state}) \vee ((q_0^4, V_4) \\ &\wedge \alpha \wedge (\Im(del(\Sigma_{ext})(\alpha)) = E^k \wedge k \geq 3) \Rightarrow \textit{safe\_state}) \end{aligned} \quad (5)$$

is an invariance in the DE-IOA model M_{rms}.

By following the model-checking technique [2], we have

Theorem 3: The predicate (5) is invariant in DE-IOA model M_{rms} once the specific state (q_{6f}^4, V_4) is not included in M_{rms}. When the specific (q_{6f}^4, V_4) appears in

Valid Channel	Internal Events	Abbr.	Fault Number
4	$e_1^4,\ e_2^4$	E_4^1	1
4	$e_{31}^4,\ e_{32}^4,\ e_{35}^4,\ e_{3f}^4, e_{61}^4,\ e_{62}^4,\ e_{63}^4,\ e_{64}^4\ e_{65}^4,\ e_{60}^4$	E_4^2	2
4	$e_{3a}^4,\ e_{3b}^4,\ e_{3c}^4,\ e_{3d}^4, e_{6a}^4,\ e_{6b}^4,\ e_{6c}^4,\ e_{6d}^4, e_{7a}^4,\ e_{7b}^4,\ e_{7c}^4,\ e_{7d}^4$	E_4^3	3
4	$e_{33}^4,\ e_{34}^4,\ e_{36}^4,\ e_{37}^4\ e_{38}^4,\ e_{39}^4,\ e_{30}^4, e_{66}^4,\ e_{67}^4,\ e_{68}^4,\ e_{69}^4$ $e_{71}^4,\ e_{72}^4,\ e_{73}^4,\ e_{74}^4, e_{75}^4,\ e_{76}^4,\ e_{77}^4,\ e_{78}^4, e_{79}^4,\ e_{70}^4,\ e_{7f}^4$	$?E_4^4$	3 or 4
4	$e_{3e}^4, e_{6e}^4, e_{7e}^4$	E_4^4	4
4	e_{3f}^4	$??E_4^4$	2 or 3 or 4
3	$e_1^3,\ e_2^3$	E_3^1	1
3	$e_{41}^3,\ e_{42}^3,\ e_{43}^3,\ e_{48}^3$	$?E_3^3$	2 or 3
3	e_{47}^3	E_3^3	3
3	$e_{44}^3,\ e_{45}^3,\ e_{46}^3$	E_3^2	2
2	e_1^2	$?E_2^2$	1 or 2

Table 1 Fault Information of the Obtained DE-IOA Model

M_{rms} model, the following predicate will be invariant instead of (5):

$$\forall \alpha \in L(M_{rms}),\ ((q_0^4, V_4) \wedge \alpha \wedge (\Im(del(\Sigma_{ext})(\alpha)) = E^k \wedge 0 \leq k \leq 1) \Rightarrow nominal\_state) \vee ((q_0^4, V_4) \wedge \alpha \wedge (\Im(del(\Sigma_{ext})(\alpha)) = E^k \wedge k \geq 2) \Rightarrow safe\_state). \quad (6)$$

Remark 4: Here (6) means the embedded RMS only has the ability to accommodate one fault within nominal operation. Actually the discrete mode q_{6f}^4 corresponds to the 2 vs. 2 case in the practical situation, since the RMS can not locate the faulty channels by majority-voting strategy and ILM information, all four signals are declared invalid by the considered RMS.

6 Conclusions

From the system engineering point of view, this paper discussed the formal verification of RMS in fault-tolerant control by using the hybrid system method. The illustration on a concrete flight RMS shows the powerful potential of hybrid system methods in practical application. However, How to improve the current RMS and systematically synthesize an efficient RMS by using the hybrid control synthesis methods will be the subjects of our future work.

Acknowledgment: The author would thank Prof. M. Blanke, Prof. Zongji Chen for many valuable discussions.

References

[1] N.R. Adam, "A New Dynamic Voting Algorithm for Distributed Database Systems", *IEEE Trans. on Knowledge and Data Engineering*, Vol.6, No.3, Jun. 1994, pp470-478.

[2] R. Alur, C. Courcoubetis, and T. A. Henzinger, "Hybrid Automata: an Algorithm Approach to the Specification and Verification of Hybrid Systems", in *Hybrid Systems, LNCS*, No.736, Springer-Verlag, 1993.

[3] S. Balemi, G. J. Hoffmann, P. Gyugyi, H. Wong-Toi and G. F. Franklin, "Supervisory Control of a Rapid Thermal Multiprocessors", *IEEE Trans. AC*, Vol.38, No.7, Jul. 1993, pp.1040-1059.

[4] M. Blanke, R. Izadi-Zamanabadi, S.A. Bogh and C. Lunau, "Fault-Tolerant Control Systems - A Holistic View", *Control Engineering Practice*, Vol.5, No.5, 1997, pp.693-702.

[5] J. Gao and Q. Xu, "Rigorous Design of a Fault Diagnosis and Isolation Algorithm", *LNCS 1567*, 1998, pp.100-121.

[6] M. Lemmon and P.J. Antsaklis, "Inductively Inferring Valid Logical Models of Continuous-state Dynamical Systems", *Theoretical Computer Science*, 138(1995) 201-210.

[7] C. J. Luh and B. P. Zeigler, "Abstracting Event-Based Control Models for High Autonomy Systems", *IEEE Trans. on SMC*, Vol.23, 1993, pp.42-54.

[8] J. Lunze, B. Nixdorf and J. Schroder, "Deterministic Discrete-Event Representations of Linear Continuous-Variable Systems", *Automatica*, Vol.35, No.3, 1999, pp.395-406.

[9] R.J. Patton, "Fault-Tolerant Control: The 1997 Situation",*IFAC SAFEPROCESS*, Hull UK, 1997, pp1033-1055

[10] P. J. Ramadge and W. M. Wonham, "Supervisory Control of a Class of Discrete Event Processes", *SIAM J. Contr. Optimiz.*, Vol.25, Jan. 1987, pp.206-230.

[11] Zhenyu Yang, Jian Lu and Zongji Chen, "Control Analysis, Synthesis and Verification in Hybrid Control Systems Using I/O Automata - A Case Study", in *Proc. IEEE CACSD'96*, Dearborn, USA, Sept 1996, pp38-43.