ISBN 978-981-14-4787-7
Proceedings of 2020 the 10th International Workshop on Computer Science and Engineering (WCSE 2020)
Shanghai, 19-21 June, 2020, pp. 464-473
doi: 10.18178/wcse.2020.06.068

The HSRN-based Reliability Analysis of Redundant Software and Hardware Integrated Systems

Xuan Hu, Jie Liu [+]

Information Security Research Center, CEPREI,
Key laboratory of Ministry of Industry and Information Technology, China

Abstract. In the areas with high reliability requirements, the use of redundancy technology can increase multiple resources of hardware and software, and achieve reasonable management of resources, thereby improving product and system reliability. In this paper, the reliability model of redundant software and hardware integrated systems is established and the reliability analysis is performed. By constructing the hierarchical stochastic reward nets model of software and hardware integrated systems, the equivalent transformation of lower-level reliability model can be realized, and the establishment process of upper-level model can hide the internal structure of the lower-level model. In this way, the upper-level model is concentrated on the use of redundant methods, which can make the upper-level model clearer. Meanwhile, the hierarchical modelling process can reduce the state space during the reliability analysis of the upper-level model, and eliminate the need to repeat the design of the same subnet. Furthermore, taking the reliability analysis of flight control computer system as an example, the feasibility and effectiveness of the method are verified, and it also provides a basis for the reliability design of similar systems.

Keywords: software and hardware integrated systems, reliability, redundancy technology, hierarchical stochastic reward nets, flight control computer

1. Introduction

Redundancy is a design method that requires two or more failures rather than a single failure to cause a predetermined undesired working condition [1]. Moreover, redundancy technology refers to the use of two or more of the same components or systems to meet the requirements of reliability and fault tolerance, to complete the same task correctly and coordinately [2]. In the areas with high reliability requirements, the systems operate independently on their own channels. If the failure rate of each single channel of a system is R, according to reliability theory, the number of channels (the number of redundancy) N and the maximum failure rate Q satisfy the following relationship: $Q = 1\text{-}R^n$, which can greatly improve the safety factor for corresponding task. The redundancy technology includes, hardware, software, and time redundancy [3]. The forms of redundancy technology are: similar redundancy, non-similar redundancy, and the working mode of redundancy channel (hot backup, cold backup) [3]. The non-similarity redundancy can greatly reduce the probability that each redundancy channel suffers a common failure and loses function at the same time, thereby enhancing the reliability of system. Thus, non-similar hardware and software technologies are often adopted in the areas with high reliability. Considering that the functions of software and hardware integrated systems (S/HIS) are completed with the support of complex hardware and software, and they also affect each other, this paper built a reliability model of redundant S/HIS from the perspective of combination of software and hardware and performed reliability analysis. Besides, the forms of redundancy of different components in a S/HIS are not the same. For a system with a redundant configuration, although the failure of partial system can cause the system to be degraded, it must be repaired in time to maintain a fault-tolerant redundant

[+] Corresponding author. Tel.: + 13671022268; fax: +020-87238359.
E-mail address: 47507346@qq.com

structure of the system. Thus, when establishing a system reliability model, the impact of maintenance and different redundancy forms on the system reliability model should be considered.

In the early days of system reliability research, simple systems were mainly used. Moreover, the number of components was small and the components were independent of each other. For this type of system, the reliability analysis methods mainly include reliability block diagram (RBD) method [4], fault tree analysis (FTA) [5], failure mode and effect analysis (FMEA) [6], etc. These methods can graphically describe the fault logic relationship between the system and the constituent units. Yet, the FTA constructs a fault tree with a large number of tasks, being prone to errors and oversight. Moreover, it is assumed that the components are independent of each other, which is not reasonable. The FMEA cannot perform quantitative analysis. The above three methods are difficult to accurately assess the system reliability during dynamic changes [7]. State space method is usually based on the Markov process [8] in reliability engineering. It constructs state transition equations and statistically analyzes system reliability indicators. However, it is suitable for systems with a small number of state spaces. When the state space is large, the readability of state transition diagram decreases, making it difficult to determine the state space and analyze the system characteristics. Monte Carlo method [9] solves system reliability problems with the help of some mathematical and technical methods generated by system probability models and random variable simulations. This method can more realistically describe the characteristics of components and systems with random properties; simple program, easy to implement, and easy to determine errors; however, disadvantages are: the convergence speed is slow during calculation, and it is generally difficult to obtain higher accuracy solutions; the error is in the sense of probability; the premise of the simulation experiment is that the input variables are independent of each other [7]. Petri nets have both graphical modelling and mathematical computing capabilities, and provides good environment for integrated modelling and analysis of complex systems. The typical representatives are Stochastic Petri Nets (SPNs) [10], Generalized Stochastic Petri Nets (GSPNs) [11], and Stochastic Reward Nets (SRNs) [12]. The SPN relates transitions with the delay of random exponential distribution. Its graphical modelling capabilities allow the model to be described intuitively. Meanwhile, the SPN can automatically generate a Continuous Time Markov Chain (CTMC), i.e., a SPN is isomorphic to a Markov chain, making it have strong mathematical calculation capabilities. The GSPN is an extension of the SPN. It divides transitions into immediate and timed transitions and is an effective way to mitigate state explosion. The SRN is an extension of SPN. It has emerged as a powerful modelling paradigm in performance, availability and reliability analysis of fault tolerant computing and communication systems as it enables the automated generation and solution of large Markov reward models [13~15]. Meanwhile, it can be considered that the SRN is a further expansion of GSPN, reflected in the system's reliability measure can be expressed in the form of rewards, and extended by adding variable cardinality arc, transition enabling function, and transition enabling priority based on the GSPN [16, 17]. In short, when the SRN is used in reliability analysis, the main advantage is that system measures can be expressed as "reward". Yet, when using the SRN to build a complex system reliability model, its state space and model readability are still affected [18]. To simplify the state space and maintain good model characteristics and readability, this paper used a Hierarchical SRN (HSRN) model [19] for system reliability analysis.

The rest of this paper is divided into the following sections, section 2 presents the model construction of HSRN. Section 3 describes the reliability modelling of flight control computer system based on HSRN. Section 4 presents case studies. Finally, section 5 shows the conclusion.

2. The Model Construction of HSRN

2.1. The Definition of SRN

Before constructing the HSRN, firstly give the basic definition of SRN [17].

Definition SRN is a 9-tuple of the form:

$$A= \{P, T, D, g, h, \mu_0, \lambda_t, \omega_t, r\} \quad (1)$$

where, 1) P is a set of places and each place can contain the tokens of non-negative integers; 2) T is a transition set and is divided into two subsets: T_t (timed transition set) and T_i (immediate transition set), $T=T_t \cup T_i$, $T_t \cap T_i=\phi$; 3) D is an arc cardinality function of marking-dependent input arc, output arc, or inhibitor arc,

i.e., the cardinalities of these arcs are variable in the SRN; 4) g is a boolean enabling function related to the identifier associated with t. When t meets the enabling conditions under the identifier μ, the enabling function $g_t(\mu)$ can be evaluated. If $g_t(\mu)$ = true, the transition t can be enabled; otherwise, t is still inhibited; 5) h is a transition enabling priority. The immediate transitions are generally specified to have a higher enabling priority than the timed transitions; 6) μ_0 is an initial identifier; 7) λ_t is an exponential distribution rate of the firing time of transition t. When the value is ∞, the enabling time of transition t is 0; ∀t∈T, if $\lambda_t(\mu) = \infty$, the identifier μ is called a virtual identifier; otherwise, μ is a true identifier. Thus, the identifier is divided into two sets of true identifier-set and virtual identifier-set, respectively denoted by Ω_T and Ω_V. $\mu \xrightarrow{t}$ represents that the transition t is enabled under the identifier μ; and under the true identifier μ, the enabling probability of transition t is: $\lambda_t(\mu)/\sum_{y\in T:\mu\xrightarrow{y}}\lambda_y(\mu)$, 8) ω_t is the enabling cardinality of transition t. Under the virtual identifier μ, the enabling probability of transition t is: $\omega_t(\mu)/\sum_{y\in T:\mu\xrightarrow{y}}\omega_y(\mu)$; 9) r is a reward rate function of identifier.

2.2. Hierarchical SRN-HSRN

The transition in the SRN can be divided into: timed transition and immediate transition [16]. The enabling time of timed and immediate transition are exponential distribution and 0, respectively. The timed and immediate transition are represented by "▯" and "▮", respectively. According to whether or not the immediate transition is enabled, the SRN divides a reachable set into two disjoint subsets: the true identifier set Ω_T that enables the timed transition and the virtual identifier set Ω_V that enables the immediate transition.

The HSRN in this study adopts a hierarchical structure that is gradually comprehensively replaced from the bottom up [19]. A lower-level model (sub-model) is used to describe the reliability of component with the failure mode being statistically independent. The lower-level model can be established through the traditional SRN. The statistical dependency of components is expressed in an upper-level model, i.e., the upper-level model is used to describe the interaction relationship of lower-level components. Thus, the lower-level model needs to be described in the upper-level model by equivalent transformation. Due to the introduction of the lower-level equivalent model, the SRN transition used to establish the upper-level model has been expanded: the timed transition set is divided into a basic and an equivalent timed transition set. The basic timed transition set is the timed transition set in the traditional SRN, and the equivalent timed transition set is used to describe the equivalent timed transition in the lower-level model, expressed by "▯". Moreover, its firing rate is the equivalent transition rate in the lower-level model.

2.3. The Measurement of HSRN

An important property of the SRN is that it can assign a reward rate to each true identifier in the Ω_T. Similarly, the model index established by the HSRN can also be expressed by the reward rate function. Let $Z(t)=r_{X(t)}$ be the immediate reward rate of system at the time t, $\pi_i(t)$ is the state probability of the true identifier i at the time t, then the expectation of immediate reward rate at the time t is: $E[Z(t)] = \sum_{i\in\Omega_T} r_i\pi_i(t)$. The expectation of steady reward rate of system is: $\lim_{t\to\infty} E[Z(t)] = E[Z] = \sum_{i\in\Omega_T} r_i\pi_i$.

When calculating system availability, the identifiers in Ω_T can be divided into normal state identifier set and fault state identifier set, which are respectively denoted as Ω_U and Ω_D. Let the reward rate function be:

$$r_i = \begin{cases} 1 & i \in \Omega_U \\ 0 & i \in \Omega_D \end{cases} \tag{2}$$

The calculation formula for the steady state availability of system is: $A_{SS} = E[Z] = \sum_{i\in\Omega_T} r_i\pi_i$.

2.4. The Equivalent Transformation of HSRN

The CTMC is constructed from the true identifier of SRN [10]. The Markov process can be used to equivalently transform the CTMC (denoted as $CTMC_L$) of low-level model based on the SRN (denoted as SRN_L). The CTMC after the equivalent transformation is recorded as $CTMC_{EL}$, and its state transition diagram is shown in Fig. 1.

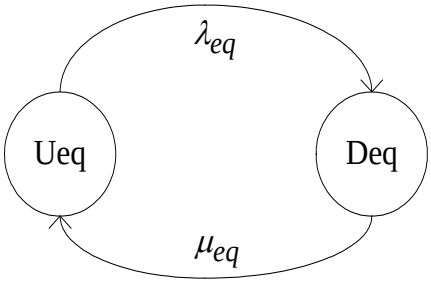

Fig. 1: The state transition diagram of $CTMC_{EL}$

The state U_{eq} and state D_{eq} in the figure indicate that the component is in a normal and fault state, and correspond to all identifiers in Ω_U and Ω_D in SRN_L, respectively. The equivalent failure rate from the state U_{eq} to D_{eq} is recorded as λ_{eq}, and the equivalent maintenance rate from the state D_{eq} to U_{eq} is recorded as μ_{eq}. The steady state probability of CTMC can be solved by the following formulas:

$$\begin{cases} \pi \cdot Q = 0 \\ \sum_j \pi_j = 1 \end{cases} \tag{3}$$

where, π is a steady state probability vector and Q is a state transition matrix. For Fig. 1: $\pi = [\pi_{U_{eq}}, \pi_{D_{eq}}]$, $Q = \begin{bmatrix} -\lambda_{eq} & \lambda_{eq} \\ \mu_{eq} & -\mu_{eq} \end{bmatrix}$. Then, the steady state probabilities of the components in $CTMC_{EL}$ in normal and fault states are: $\pi_{U_{eq}} = \mu_{eq}/(\lambda_{eq} + \mu_{eq})$, $\pi_{D_{eq}} = \lambda_{eq}/(\lambda_{eq} + \mu_{eq})$. The formulas for calculating the equivalent transition rate in $CTMC_{EL}$ by $CTMC_L$ [19] are further given: $\lambda_{eq} = \sum_{t_{i,j} \in F} \Pr\{s_i|U\} \cdot q_{i,j} = (\sum_{t_{i,j} \in F} \pi_i \cdot q_{i,j}) / \sum_{s_k \in U} \pi_k$, $\mu_{eq} = \sum_{t_{i,j} \in M} \Pr\{s_i|D\} \cdot q_{i,j} = (\sum_{t_{i,j} \in M} \pi_i \cdot q_{i,j}) / \sum_{s_k \in D} \pi_k$. S_i is the state i in $CTMC_L$; $t_{i,j}$ and $q_{i,j}$ are the transition and transition rate from the state i to j; π_i is the steady state probability of state i; U and D are the set of normal state and fault state respectively.

Further referring to reference [19], $\pi_{U_{eq}} = \pi_U$, $\pi_{D_{eq}} = \pi_D$, where, $\pi_U = \sum_{s_k \in U} \pi_k$, $\pi_D = \sum_{s_k \in D} \pi_k$.

This shows that the probability that a component is in a normal state (fault state) in $CTMC_{EL}$ is equal to the sum of the probability in a normal state set (fault state set) in $CTMC_L$, which guarantees the equivalence in the transformation process when the $CTMC_L$ is replaced with the $CTMC_{EL}$. Thus, the equivalent stochastic reward net (SRN_{EL}) shown in Fig. 2 that is isomorphic to the $CTMC_{EL}$ and SRN_L are also equivalent. The place P_u and P_d in Fig. 2 represent the normal and fault state of component respectively. Moreover, the activation rates of the transitions T_f and T_r are equivalent transition rates λ_{eq} and μ_{eq} respectively, thereby realizing the equivalent transformation of the underlying SRN model.

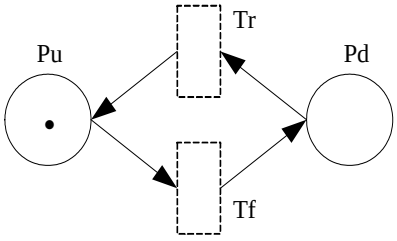

Fig. 2: The schematic diagram of SRN_{EL}

2.5. The Model Solving for HSRN

The SRN hierarchical process ensures the firing time of the equivalent transition obeys an exponential distribution; thus, the stochastic process constructed by the true identifier of HSRN is also a CTMC.

The state transition matrix of CTMC can be obtained by calculating the transition rate between true identifiers. Define $P_V = [P_{VV} \mid P_{VT}]$ as the transition probability matrix from a virtual identifier to virtual identifier (P_{VV}) or true identifier (P_{VT}), and $U_T = [U_{TV} \mid U_{TT}]$ as the transition probability matrix from a true identifier to virtual identifier (U_{TV}) or true identifier (U_{TT}), the transition matrix between true identifiers is [17]:

$$U = U_{TT} + U_{TV}(1 - P_{VV})^{-1} P_{VT} \tag{4}$$

Define the state transition matrix of the CTMC as Q ($Q_{i,j}$ is the transition rate from state i to j), then the

elements in the system state transition matrix are:

$$Q_{i,j} = \begin{cases} U_{i,j} & i \neq j \\ -\sum_{k \in \Omega, k \neq i} U_{i,k} & i = j \end{cases} \quad (5)$$

Let π be the steady state probability vector corresponding to the true identifier. Based on formulas (3) and (5), the steady state probability of each state in the CTMC can be calculated and the system reliability analysis can be performed.

3. The Reliability Modelling of Flight Control Computer System Based on HSRN

3.1. The Function and Structure of Flight Control Computer System

Flight control computer is the core of aircraft. Moreover, its high reliability is the key to the high reliability of flight control system and also the guarantee of the safe flight of aircraft. Thus, the redundancy technology is usually used to design the flight control computer. The essence is to increase the reliability of flight control computer by increasing redundancy resources and shielding the influence of faulty components. This section selected a certain type of aircraft flying-by-wire main flight control computer for research.

Fig. 3 shows the structure of the flying-by-wire main flight control computer for a certain type of aircraft. It is a non-similar dual redundancy system. All the redundancy hardware devices are divided into left and right groups, i.e., dual channels. The main flight control computer system sends instructions to servo actuators. The servo actuators receive control instructions and send signals to steering gears to drive the corresponding rudder surface to change the attitude and speed of aircraft. Each channel has three non-similar hardware and software branches, and bus communication is used between each channel. The input and output part of the branch include two bus terminals, one for receiving and the other for transmitting / receiving. Each channel receives inputs from two groups of buses, but only transmits data to the same group of buses. When one group of buses fails, it does not affect the normal work of the other group. The three branches of the main flight control computer system are respectively assigned to instruction branch, backup branch and monitoring branch. The instruction branch solves control law and transmits instructions to the designated bus. The other two branches perform monitoring function and branch redundancy management task, respectively. Once the instruction or monitoring branch fails, its task is replaced by the backup branch. Fault of any of the other two branches will cause the system output to be disconnected.

3.2. The System Reliability Modelling Based on HSRN

In this section, the HSRN was used to perform the reliability modelling and analysis in the normal state of a certain aircraft flying-by-wire main flight control computer. The digital main flight control computer is responsible for performing control law calculations and is composed of multiple branches and channels in parallel. Analyzing the overall structure and function of the system, the types of system components include power supply, CPU, I / O interface, software, and bus. The hardware of the three branches in each channel

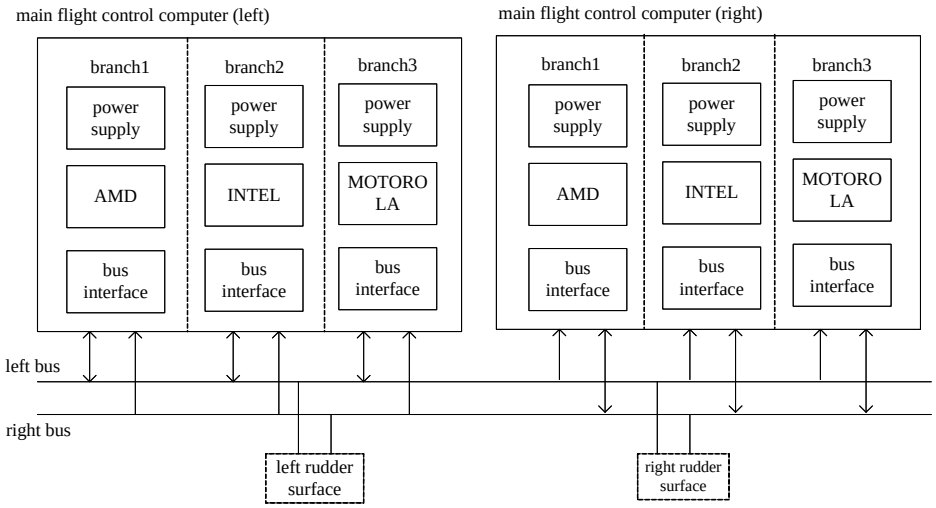

Fig. 3: The structure of a certain type of aircraft flying-by-wire main flight control computer

are not similar, and the software are also not similar; each channel is powered by the corresponding left, middle, and right power bus; each channel receives data from two buses, and only sends data to the corresponding one bus. The aircraft's flying-by-wire main flight control computer system has a parallel relationship between the left main control computer and the right main control computer; the three branches of the main control computer have a parallel relationship; the components in a single branch have a series relationship. Moreover, from the system perspective, the main flight control computer system and the bus are connected in series. The modelling process assumes that the system hardware and software obey an exponential distribution with a constant failure rate, and does not consider the immediate fault. When the main flight control computer fails during the flight, maintenance conditions are generally not available; therefore, the system can be modelled as a non-repairable system. In the case of ground maintenance, the flight control computer needs to be studied as a repairable system. Thus, the reliability model construction needs to be considered in two cases.

1) Repairable

This study analyzed hardware and software as a whole. The following considers equipment hardware (including software) / component failures, maintenance, and equipment redundancy. Moreover, the HSRN is adopted to establish the reliability model of branch, main control computer, and main flight control computer system. Considering that the software of main flight control system is closely related to the hardware equipment, this study analyzed the software and chip of the main flight control system as a whole.

(1) The reliability model of branch

In the branch reliability model shown in Fig. 4, the model composed of the place $P_{x.u}$, $P_{x.d}$ and the transition $t_{x.f}$, $t_{x.r}$ between the places is used to describe the failure and maintenance of branch power, chip and bus interface. The place $P_{x.u}$ and $P_{x.d}$ respectively indicate the normal state and fault state of the component x, and the transitions $t_{x.f}$ and $t_{x.r}$ are used to describe the fault and maintenance process of the component x. Their firing rates are recorded as λ_x and μ_x (λ_x and μ_x are the fault rate and maintenance rate of branch component x respectively). The interaction between the branch software / hardware is as follows:

- The boolean enabling function of the transition t1 and t2 is $((\#(P_{p.u})=0) \cup (\#(P_{b.u})=0))$.

This is because the fault of the power supply or the fault of the bus in the branch will cause the running chip to stop (not to fail). When the transition $t_{p.r}$ or $t_{b.r}$ occurs, a token will be placed in the place $P_{c.u}$. This is because after the maintenance of the power supply or bus is completed, with the restart of the power supply or bus, the chip that can run under its normal working condition can be re-run.

- The boolean enabling function of the transition t3 and t4 is $\#(P_{p.u})=0$.

This is because the fault of the power supply in the branch will stop the running bus (not a failure). When the transition $t_{p.r}$ occurs, a token will be placed in the place $P_{b.u}$. This is because after the maintenance of the power supply is completed, with the restart of the power supply, the bus that can be operated under its normal working condition can be re-run.

A CTMC isomorphic to the branch reliability model of Fig. 4 is shown in Fig. 5.

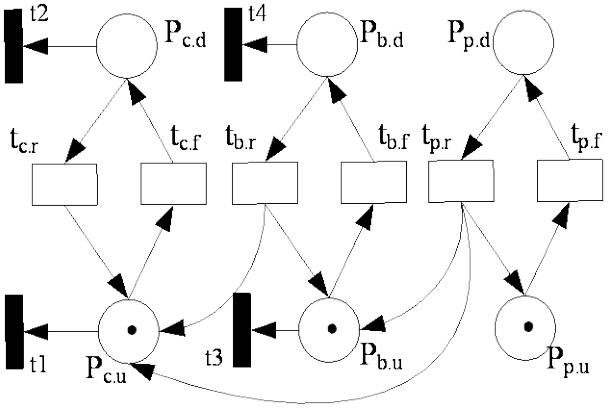

Fig. 4: The reliability model of branch

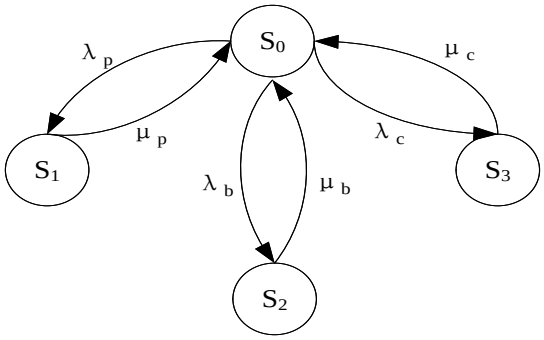

Fig. 5: The isomorphic CTMC of branch reliability model

The distribution of the tokens corresponding to the states $S_0 \sim S_3$ in Fig. 5 is shown in Tab. 1.

Table. 1: The token distribution of branch reliability model

	$P_{p.u}$	$P_{p.d}$	$P_{b.u}$	$P_{b.d}$	$P_{c.u}$	$P_{c.d}$
S_0	1	0	1	0	1	0
S_1	0	1	0	0	0	0
S_2	1	0	0	1	0	0
S_3	1	0	1	0	0	1

There are four states in the CTMC of the branch reliability model shown in Fig. 5. iff, $\#(P_{c.u})=1$, the branch is in a normal state, otherwise it is a fault state. Thus, there are one normal state and three fault states, i.e., the normal state set is $\Omega_U = \{S_0\}$, and the fault state set is $\Omega_D = \{S_1, S_2, S_3\}$. Let π be the steady state probability vector and Q be the state transition matrix. The state transition matrix Q can be obtained as:

$$Q = \begin{bmatrix} -(\lambda_p+\lambda_b+\lambda_c) & \lambda_p & \lambda_b & \lambda_c \\ \mu_p & -\mu_p & 0 & 0 \\ \mu_b & 0 & -\mu_b & 0 \\ \mu_c & 0 & 0 & -\mu_c \end{bmatrix} \quad (6)$$

The element $q_{i,j}$ in the matrix Q is the transition rate from state S_i to S_j. When the branch failure and maintenance parameters are given, the steady state probability π_i (i = 0,1,2,3) of the branch in the state S_i can be calculated according to formula (3). Moreover, the equivalent failure and maintenance rate of the branch are: $\lambda_{eq\_ws} = (\pi_0 \times q_{0,1}+\pi_0 \times q_{0,2} + \pi_0 \times q_{0,3})/\pi_0$, $\mu_{eq\_ws} = (\pi_1 \times q_{1,0}+\pi_2 \times q_{2,0} + \pi_3 \times q_{3,0})/(\pi_1 + \pi_2 + \pi_3)$.

(2) The reliability model of main control computer system (MCCS) (left / right)

When the equivalent transition rate of a branch is solved, the reliability model of the left (right) MCCS can be established according to the parallel operation mode and the number of parallel connections of the branch, as shown in Fig. 6. The places $P_{cs.u}$ and $P_{cs.d}$ in Fig. 6 indicate that the left (right) MCCS works normally and fails. The place $P_{i.u}$ (i = 1,2,…, n, here, n = 3) and $P_{i.d}$ represent the state of the i^{th} branch of the left (right) main control computer system, respectively. The transition $t_{i.f}$ and $t_{i.r}$ are used to describe the failure and maintenance process of the i^{th} branch, whose firing rate is the equivalent failure rate λ_{eq_cs} and equivalent maintenance rate μ_{eq_cs} of a single branch. Due to space limitations, the solution process of the left (right) MCCS reliability model is no longer listed here.

(3) The reliability model of MCCS (as a whole)

After the equivalent transition rate of left (right) MCCS is solved, the total reliability model of MCCS can be established according to parallel operation mode and the number of parallel connections. The process is similar to the previous step, and not repeated here.

(4) The reliability model of system

After solving the equivalent transition rate of the reliability model of the main control computer system, since the system and external bus are connected according to the serial working mode, the final system reliability model can be established as shown in Fig. 7.

In Fig. 7, the subnets formed by the place $P_{x.u}$ (x = cs, Bs, st, the same below) and $P_{x.d}$ and the transitions $t_{x.f}$, $t_{x.r}$ between the places are the equivalent reliability models of the MCCS (x = cs) and bus system (x = Bs), respectively. The firing rates of $t_{x.f}$ and $t_{x.r}$ are the equivalent transition rates of the underlying reliability

models of the corresponding subsystems; the place $P_{st.u}$ and $P_{st.d}$ are used to represent the state of entire system. According to the system reliability model shown in Fig. 7, using the equivalent failure rate and equivalent maintenance rate of the equivalent reliability model of the MCCS and bus system, the equivalent transition rate of entire system can also be solved.

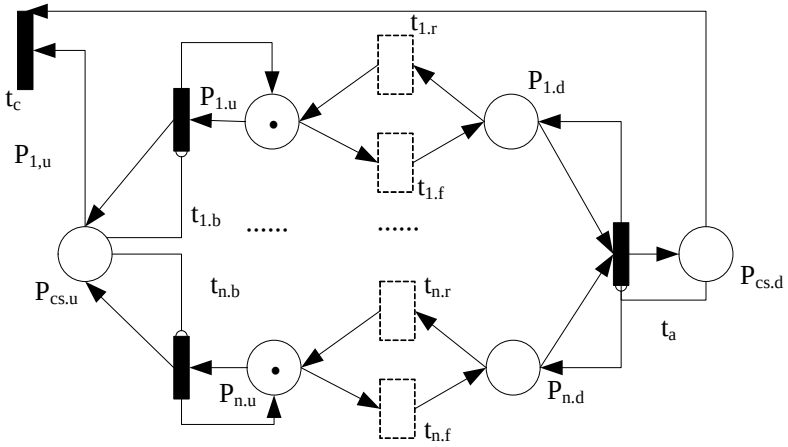

Fig. 6: The reliability model of MCCS (left / right) $P_{1,u}$

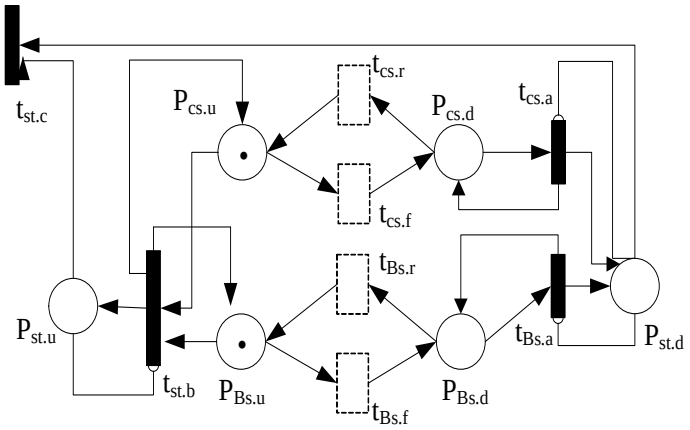

Fig. 7: The reliability model of system (repairable)

2) Non-repairable

In the case of non-repairable, the RBD method can be used to model the system according to functional logic dependencies, as shown in Fig. 8. The form of Fig. 8 is a typical hybrid system. A hybrid system refers to a system composed of a series system and a parallel system. Let the reliability of each unit of each parallel system in the parallel-serial system in the figure be R_{ij} (t), (i = 1,…, 3; j = 1,…, 6) and R_{bus}. Then the reliability of the first part and second part of the parallel system are as formula (7) and (8), respectively:

$$R_1(t) = 1 - \prod_{i=1}^{3}[1 - \prod_{j=1}^{3} R_{ij}(t)] \quad (7) \qquad R_2(t) = 1 - \prod_{i=1}^{3}[1 - \prod_{j=4}^{6} R_{ij}(t)] \quad (8)$$

Then use series system calculation formula to calculate the reliability of parallel-series system:

$$R(t) = \{1 - \prod_{i=1}^{3}[1 - \prod_{j=1}^{3} R_{ij}(t)]\} * \{1 - \prod_{i=1}^{3}[1 - \prod_{j=4}^{6} R_{ij}(t)]\} * R_{bus} \quad (9)$$

With the system reliability, other system reliability features can be calculated accordingly.

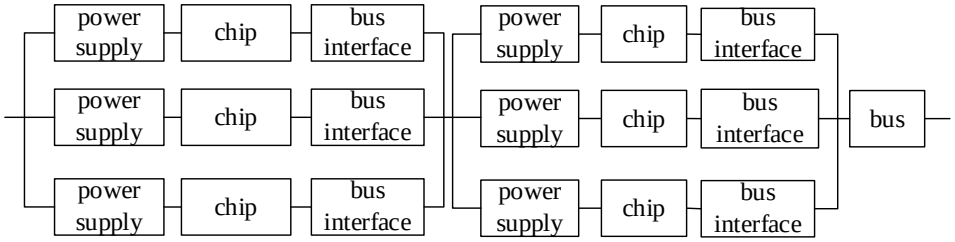

Fig. 8: The reliability model of system (non-repairable)

4. Case studies

4.1. Parameter Setting

This section took the reliability analysis of a certain type of aircraft flying-by-wire main flight control computer system as an example, and the system was modelled with the HSRN to obtain the system reliability index. Its structure is shown in Fig. 3. The reliability parameters of the hardware and software components of equipment in subsystem are given in Tab. 2 according to system design requirements and actual engineering [20, 21]. It should be noted that, because the situation of non-repairable is relatively simple, it is omitted here.

Table. 2: The failure and maintenance parameters of component of a certain type of aircraft flying-by-wire main flight control computer system

No.	component	failure rate(/h)	maintenance rate(/h)
1	power supply	2.0×10^{-6}	0.5
2	chip	1.0×10^{-6}	2
3	bus interface	2.0×10^{-6}	0.5

4.2. Reliability Analysis

Calculate the equivalent transition rates of the corresponding branch, subsystem, and system according to the failure and maintenance parameters provided in Tab. 2. The calculation process of the equivalent transition rates of branch is given below. According to the component failure and maintenance parameters shown in Tab. 2, $\lambda_p = 2.0 \times 10^{-6}/\mathrm{h}$, $\mu_p = 0.5/\mathrm{h}$; $\lambda_b = 1.0 \times 10^{-6}/\mathrm{h}$, $\mu_b = 2.0/\mathrm{h}$; $\lambda_c = 2.0 \times 10^{-6}/\mathrm{h}$, $\mu_c = 0.5/\mathrm{h}$. Combining formula (6) and (3), the steady state probability vector of the reliability model of a single branch can be calculated as: $\pi = [0.99999 \;\; 4.0 \times 10^{-6} \;\; 4.9999 \times 10^{-7} \;\; 4.0 \times 10^{-6}]$, where, π_i (i = 0,1,2,3) is the steady state probability when a single branch is in the state S_i. Calculate the equivalent failure rate and equivalent maintenance rate of a single branch: $\lambda_{eq\_ws} = (\pi_0 \times q_{0,1} + \pi_0 \times q_{0,2} + \pi_0 \times q_{0,3})/\pi_0 = 5.0 \times 10^{-6}/\mathrm{h}$, $\mu_{eq\_ws} = (\pi_1 \times q_{1,0} + \pi_2 \times q_{2,0} + \pi_3 \times q_{3,0})/(\pi_1 + \pi_2 + \pi_3) = 0.5882/\mathrm{h}$.

To verify the feasibility and accuracy of the HSRN method used in this paper, the steady state probability of a single branch in normal operation can be numerically calculated. The equivalent model was used to calculate the steady state probability of a single branch in the normal state. Let the steady state probability vector of a single branch be: $\pi = [\pi_{Ueq\_ws} \;\; \pi_{Deq\_ws}]$. The state transition matrix of a single branch equivalent model is: $Q = \begin{bmatrix} -\lambda_{eq_{ws}} & \lambda_{eq_{ws}} \\ \mu_{eq_{ws}} & -\mu_{eq_{ws}} \end{bmatrix}$. The steady state probability that a single branch is in the normal state can be solved as: $\pi_{Ueq\_ws} = \mu_{eq_{ws}}/(\lambda_{eq_{ws}} + \mu_{eq_{ws}}) = 0.99999$.

Next, the steady state probability of a single branch in the normal state was calculated without using the equivalent model. The normal state identifier set is $\Omega_U = \{S_0\}$ and the fault state identifier set is $\Omega_D = \{S_1, S_2, S_3\}$. Let the reward rate $r_0 = 1$, $r_1 = r_2 = r_3 = 0$, then the steady state probability that a single branch is in the normal state is: $\pi_{U_{ws}} = \sum_{i \in \Omega_T} r_i \pi_i = \pi_0 = 0.99999$. $\pi_{Ueq\_ws} = \pi_{U_{ws}}$ indicates that the calculation results of the steady state probability of a single branch in the normal state are the same, thereby verifying the feasibility of the equivalent transformation of the underlying model used in the paper. It can be seen that the HSRN method can provide an effective method for simplifying the state space of system reliability model. Furthermore, the equivalent failure rate and equivalent maintenance rate of entire system can be obtained. Finally, the system's availability indicators, MTBF and other reliability indicators can be obtained. Due to space limitations, the specific process is omitted here.

5. Conclusion

This paper studied the reliability modelling and analysis of the redundant S/HIS, considering the maintenance process of system. The HSRN method was adopted to analyze the reliability of system. Through the equivalent transformation of the lower-level reliability model, the establishment process of upper-level model can hide the internal structure of the lower-level model, i.e., through the equivalent transformation of the reliability model of a single branch, the interaction between the lower-level hardware and software components and the failure and maintenance processes of the components need not to be

considered again when establishing the upper-level reliability model of system. In this way, the upper-level model is concentrated on the use of redundant methods, which can make the upper-level model clearer. Meanwhile, the hierarchical modelling process can reduce the state space during the reliability analysis of the upper-layer model, and eliminate the need to repeat the design of the same subnet. The HSRN provides a practical method for the reliability analysis of redundant S/HIS.

6. References

[1] MIL-F-9490D, Military specification, Flight control systems-design, installation and test of piloted aircraft, general specification for, 1975.

[2] Committee for terms in aviation sciences and technologies. Chinese terms in aviation science and technology. *Science Press*, 2004.

[3] Ma Qiuyu. The real-time redundancy software design for UAV flight control system. Xi'an: Northwestern Polytechnical University, 2007.

[4] Guo H. T., Yang X. H. A simple reliability block diagram method for safety integrity verification. *Reliability Engineering & System Safety*, 2007, 92(9): 1267-1273.

[5] Lee W. S., Grosh D. L., Tillman F. A., et al. Fault tree analysis, methods and applications-a review. *IEEE Transaction on Reliability*, 1985, 34(3): 194-203.

[6] Stamatis D. H. Failure mode and effect analysis: FMEA from theory to execution. *AsqPress*, 2003.

[7] Jia Limin, Lin Shuai. Current status and prospect for the methods of system reliability. *Systems Engineering and Electronics*, 2015, 37(12): 2887-2893.

[8] Pukite P., Pukite J. Markov modeling for reliability analysis. *Wiley-IEEE Press*, 1998.

[9] Metropolis N., Ulam S. The Monte Carlo method. *Journal of the American Statistical Association*, 1949, 44(247): 335-341.

[10] Ciardo G., Muppala J., Trivedi K. S. SPNP: stochastic Petri net package. *Proceedings of the 3rd International Workshop on Petri Nets and Performance Models*, 1989: pp. 55-60.

[11] Constantinescu C. Dependability evaluation of a fault-tolerant processor by GSPN modeling. *IEEE Transactions on Reliability*, 2005, 54(3): 468-474.

[12] Ungsunan P. D., Chuang L., Yang W., et al. Network processing performability evaluation on heterogeneous reliability multicore processors using SRN model. *Proceedings of the IEEE International Conference on Parallel & Distributed Processing*, 2009: pp. 1-6.

[13] F. Longo, et al. A scalable availability model for infrastructure-as-a-service cloud. *Proceedings of the IEEE/IFIP 41st International Conference on Dependable Systems & Networks*, 2011: pp. 335-346.

[14] R. Ghosh, F. Longo, V. K. Naik, and K. S. Trivedi. Modeling and performance analysis of large-scale IaaS clouds. *Future Generation Computer Systems*, 2013, 29(5): 1216-1234.

[15] Reza Entezari-Maleki, Kishor S. Trivedi, Ali Movaghar. Performability Evaluation of Grid Environments Using Stochastic Reward Nets. *IEEE Transactions on dependable and secure computing*, 2015, 12(2): 204-216.

[16] Lin Chuang. Stochastic Petri Nets and System Performance Evaluation. *Tsinghua University Press*, 2005.

[17] Tomek L., Trivedi K. S. Analyses using stochastic reward nets. *John Wiley & Sons*, 1995.

[18] Gianfranco Ciardo, Trivedi K. S. A Decomposition approach for stochastic reward net models. *Performance Evaluation*, 1993, 18(1): 37-59.

[19] Yu Min, He Zhengyou, Qian Qingquan. Reliability Analysis on Integrated Supervision & Control System in Metro based on HSRN. *Journal of the China Railway Society*, 2012, 34(2): 70-79.

[20] Jiang Yinqing. Flight Control System Software Design for Small-Scale UAVs based on μC/OS-II. Nanjing: Nanjing University of aeronautics and astronautics, 2009.

[21] Han Guotai. Prognostics and Health Management of Avionics. *Avionics Technology*, 2009, 40(1): 30-38.